For The Serious User Of Apple ][ Computers

# COMPUTIST

Issue No. 46 August 1987

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

Softkeys For:

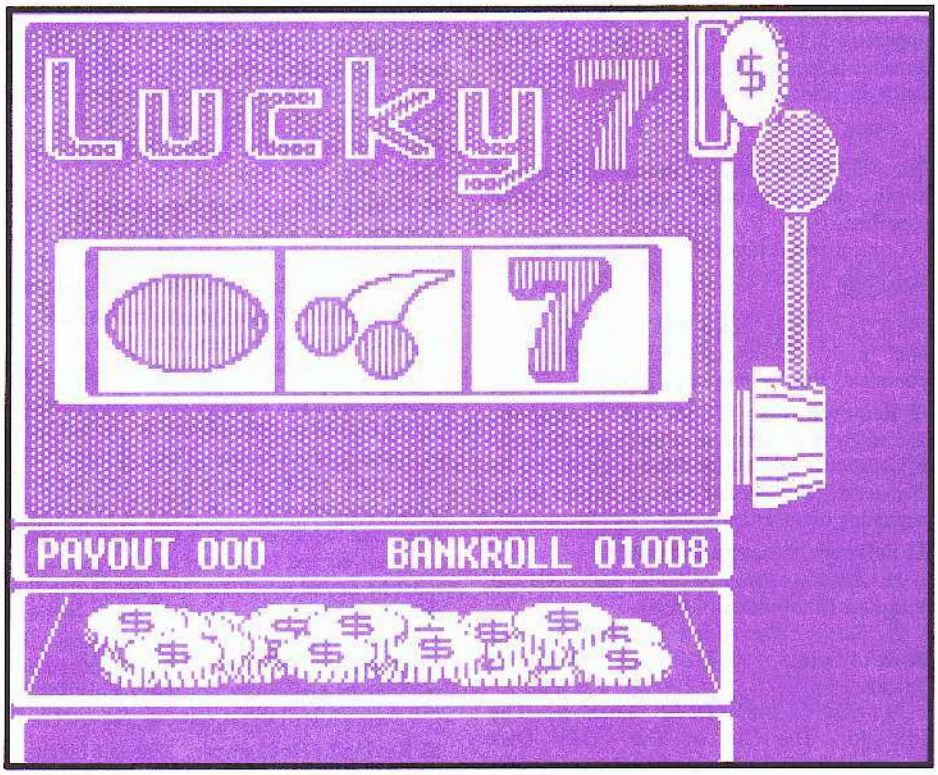
## Video Vegas
## Star Blazer
## Science Toolkit
## The Handlers
## Law of the West
## OGRE
## Puzzles and Posters

Feature:

## Amazing
## Computer Facts

Core:

## Shape Magic




(Page 7)

COMPUTIST
PO Box 110846-T
Tacoma, WA 98411

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

# Coping With COMPUTIST

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

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

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

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

**PR#6**

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

Control characters are specially boxed:

**6[CTRL-P]**

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

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

■ **Software Recommendations:**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**10HOME:REMCLEAR SCREEN**

The LIST will look like:

**10 HOME : REM CLEAR SCREEN**

because Applesoft inserts spaces into a program listing before and after every command word or mathematical operator. These spaces usually don't pose a problem except in line numbers which contain REM or DATA commands. There are two types of spaces: those that have to be keyed and those that don't. Spaces that must be keyed in appear in COMPUTIST as delta characters (Δ). 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 (Δ) spaces after DATA statements.

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

**CALL -151**

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

When finished, return to BASIC with:

**E003G**

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

■ **Keying In Source Code** The source code is printed to help explain a program's operation. To key it in, you will need the S-C Assembler. Without this assembler, you will have to translate pieces of the source code into something *your* assembler will understand. A table of S-C Assembler directives appears in COMPUTIST 17.

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

■ **CHECKSOFT Instructions:**

**LOAD** *filename*
**BRUNCHECKSOFT**

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

■ **CHECKBIN Instructions:**

**CALL -151**
**BLOAD** *program filename*

Install CHECKBIN at an out of the way place

**BRUN CHECKBIN,A$6000**

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

**xxx.xxx[CTRL-Y]**

Correct the lines at which the checksums differ.

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

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

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

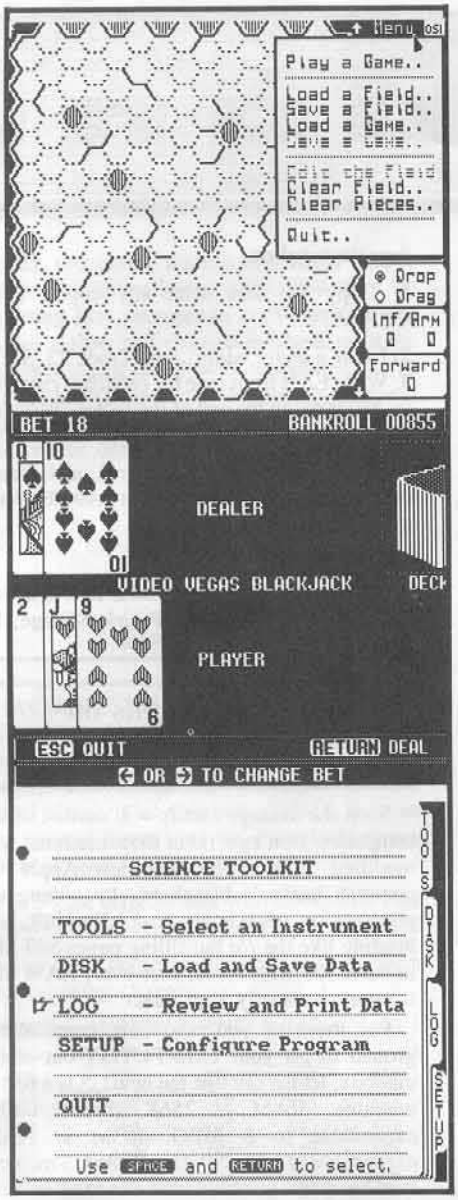
..."It is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or ***adaptation*** of that computer program provided:

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

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

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

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




*This month's cover:*
*Graphics from Baudville's "Video Vegas."*

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

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

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

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

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

Apple usually refers to an Apple ][ 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.

# COMPUTIST

Issue 46 August 1987

**Publisher/Editor:** Charles R. Haight **Printing:** Valco Graphics, Seattle WA
**Editorial Phone:** (206) 474-5750 **Advertising phone:** (206) 474-5750

## softkeys:

**12** Advanced Microsystems Technology Programs
*by Jim S. Hart*

**13** Word Attack revisited
*by Jim S. Hart*

**24** Star Blazer
*by Rich Etarip*

**26** Science Toolkit
*by Stephen Lau*

## features:

**14** The Shift Key and the Lowercase Option for the ][ Plus
Two features missing from the Apple ][ plus (which are present on the //e and //c) are a real shift key and the ability to display lowercase text. This article shows how you can overcome these shortcomings. *by Gary Kowalski*

**22** Amazing Computer Facts
This article presents some little known and very interesting information about your computer. *by Rich Etarip*

## core:

**16** Multiscribe: a review
Everything you've always wanted to know about this word processing program. *by Bruce Mager*

**18** Shape Magic
If you are an Applesoft programmer who is tired of coding shape tables by hand, this handy utility program is for you. *by Burton Lo*

## departments:

**4** Input

**7** Readers' Softkey & Copy Exchange

Softkeys for: Broderbund's **The Color Enhanced Print Shop** *by Jason Rosenwald*, Baudville's **Video Vegas** *by Lyle Marentette*, Advanced Logic Systems' **The Handlers** *by Jason Lee*, Richard D. Irwin's **K.C. Deals On Wheels** *by Jim S. Hart*, Accolade's **Law of the West** *by The Guildmaster*, Gentry's **Break the Bank Blackjack** *by Jim S. Hart*, DCH Educational's **Foundation course in Spanish** *by Jim S. Hart*, Origin's **OGRE** *by Charles Taylor*, MECC's **Puzzles and Posters** *by Paul Giguere*

# input

**Please address letters to:**

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

Include your name, address and phone number.

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

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

## Spellworks v1.6

Recently, I was reading your magazine to find out how to deprotect Spellworks. Unfortunately, the directions were for version 1.5, and I have version 1.6. However, the only difference between the two versions is where the protection code starts. The following directions should be used if you have version 1.6.

```
UNLOCK SPELL.SYSTEM
BLOAD
SPELL..SYSTEM,TSYS,A$2000
CALL -151
2097.2099
```

At this point, if you have version 1.6 you will have the values 20 00 BF

```
2097: 4C FA 20
<ctrl-C>
BSAVE
SPELL.SYSTEM,TSYS,A$2000
LOCK SPELL.SYSTEM
DELETE TMP
DELETE CP
```

And for the mailmerge side:

```
UNLOCK MERGE.SYSTEM
BLOAD MERGE.SYSTEM
CALL -151
204E.2050
```

At this point, if you have version 1.6 you will have the values 20 00 BF.

```
204E: 4C 7E 20
<ctrl-C>
BSAVE
MERGE.SYSTEM,TSYS,A$2000
LOCK MERGE.SYSTEM
DELETE MRGMSG
```

All of this should be done on a copy which can be made using the FILER program on the /USERS.DISK. After these instructions are done, the program can be moved to any disk including hard drives. This brings up a point, after moving the SPELL.SYSTEM and LEX files to my hard drive is there any way to change the default location of the 963LEX files that I don't have to change everytime I use the program?

George B. Murray III
Rolling Meadows, IL

## Great American Labyrinth

I am the owner of two disks from Activision that, until I received COMPUTIST No. 39, defied all attempts to copy. Thanks to your magazine, they are now both fully deprotected and safe from my children.

The first disk was Great American Cross Country Road Race, and the second was Labyrinth. COPYA was able to copy both without any errors. That indicated that a nibble count was being used. I had been unable to find it. Larry Rando's softkey in COMPUTIST No. 39 for Great American Cross Country Road Race inspired me to see if Activision was using the same routine for Labyrinth. Like most publishers, Activision will use the same protection for several disks. I was in luck, and so were other owners of Labyrinth.

The first part of the disk read routine on Labyrinth was found at track 22, sector 3, byte C2 (it begins A9 56 85.) It is slightly different than Mr. Rando's disk, in that the first two bytes of the second routine were found in the last two bytes of the first sector, rather than the first two bytes of the following sector. Covering the code from C2 to FF with EA's did nothing but make the disk grind. Searching the next sector (sector 4) did not reveal the rest of the missing code (25 FC). A quick search with my Copy ][ Plus revealed that the code was to be found on the previous sector (sector 2). Changing these to the required A9 FF resulted in a completely deprotected disk.

1) Copy the disk with any copier (even COPYA)

2) Using a sector editor, read Track 22, Sector 3. Write EA's from byte C2 (which is currently an A9) through the end of the sector. Write the changes back to your copy.

3) Now read Track 22, Sector 2, and change the first two bytes from 25 FC to A9 FF. Write it back.

4) Have fun.

John R. Nicholson
Prairie Village, KS

## Some Thoughts on a New Machine

Soon the average family will consist of 2.3 computers. Don't get rid of those machines with modified ROMs yet! The new Apple GS commits hacker's blasphemy by wiping out memory on open-apple-reset. Not just hole blasting like the //e but filling pages $08-$BF in bank $00 with zeros. However, RAM disk memory is usually kept intact.

For those of you who only come above ground to get your COMPUTIST out of the mailbox, let me say that the new GS is a 65816 machine. RAM is 256K (breadcrumb), expandable to 8 MEG. ROM is 128K, expandable to 1 MEG!! The Apple GS memory card has a nice price but little else to offer. Both the Applied Engineering GS RAM card and the AST card have more expandability including sockets for user ROM expansion (standard on the AST).

The GS offers instant access to a desktop accessory menu at almost any time by means of an Open-Apple-ESC. The desktop menu can take you to a battery backed control panel or, in the future, to other accessories which can be loaded into RAM or ROM! Perhaps some enterprising programmer will write a set of cracking utilities that can be used as a desktop accessory. Or at the very least, a way to enter the monitor from the desktop menu.

It is possible to disable the interrupts that the GS needs to enter the desktop menu. We need to speak out early against this by refusing to buy ANY software that disables the desktop menu. This kind of protection maims the potential of the GS.

Activision's Paintworks Plus is being called Paintworks Minus in some circles because of the shortcomings imposed by the protection scheme. It trashes any memory over the 512K

# input

it uses, precluding the use of a RAM disk to speed up its animation routines. BBS users are reporting that attempts to copy the disk can initiate a slow self destruct mechanism in the program. Only one disk is supplied.

I am not aware of any bit copiers that will work with the 3.5 drive. I hope COMPUTIST will run an occasional GS related article.

David Todd
Cambridge, MD

## Fight Night

In COMPUTIST No. 40, you reviewed Fight Night. Fight Night has a feature called boxing construction, which saves the boxers on the original disk. Since this is a dangerous move I tried to make a bit copy with EDD III. Without any success it was time for a crack. Here are the cookbook instructions:

1) Copy the original disk with Locksmith Fast coipy or COPYA. Ignore errors on track $22

2) Use your favorite sector editor to perform the following changes:

| TRACK | SECTOR | BYTE | FROM | TO |
|---|---|---|---|---|
| $0D | $00 | $B1 | $20 | $EA |
| $0D | $00 | $B2 | $28 | $EA |
| $0D | $00 | $B3 | $62 | $EA |
| $0D | $00 | $B4 | $A9 | $EA |
| $0D | $00 | $B5 | $00 | $EA |
| $0D | $00 | $CF | $CE | $EA |
| $0D | $00 | $D0 | $E9 | $EA |
| $0D | $00 | $D1 | $61 | $EA |
| $0D | $00 | $D2 | $30 | $EA |
| $0D | $00 | $D3 | $0A | $EA |
| $0D | $00 | $D4 | $20 | $EA |
| $0D | $00 | $D5 | $28 | $EA |
| $0D | $00 | $D6 | $62 | $EA |
| $0D | $00 | $D7 | $4C | $EA |
| $0D | $00 | $D8 | $42 | $EA |
| $0D | $00 | $D9 | $61 | $EA |

WaKit So
British Columbia, Canada

## More Silent Service

In addition to my softkey for Silent Service published in COMPUTIST No. 39, I would like to offer the following alternatives for cracking the program in case my method, or Jim Hart's method, won't work.

1) Read in track $05, sector $09, and edit byte C9 to CD. Read in track $05, sector $07, and edit byte 05 to 60 EA EA.

2) Read in track $04, sector $09, and edit byte B8 to DB. Read in track $05, sector $09, and edit byte B8 to DB.

Please note that method 2 is similar to Jim Hart's softkey, but it has an additional edit.

p.s. If anyone out there is into telecommunications, then give the Electronic Odyssey a call evenings (6pm-6am) and weekends (24 hours) at (303) 474-5795. It's based in Farmington, MI, and runs on a 20 meg hard drive.

Greg Poulos
Novi, MI

## Operation Frog

I started looking at this disk with the sector editor in Copy ][ Plus. By using the custom DOS option I saw the DATA epilogue had been altered. With Super IOB and a modified fast controller I transferred the files over to a normal DOS. At this point I put in brake points into the code to find what was being loaded where. Some code was loaded at $BB00 which normally is a nybble buffer, then relocated at $200. This code did the disk verification with two exit points. One exit was for an acceptable disk and continues the program loading, the other was for errors. By changing the jump instruction to exit at the acceptable disk exit. The controller will do everything needed to make a COPYAable copy. One more thing: Diversi-DOS and ProntoDOS didn't work but a little known fast DOS (DOS 3.4) did work to help speed the loading.

```
1000 REM FROG
1010 TK = 0 :LT = 30 :ST = 15 :LS = 15 :CD = WR
     :FAST = 1
1020 GOSUB 170 : GOSUB 490 : GOSUB 610
1025 GOSUB 310
1030 GOSUB 230 : GOSUB 490 : GOSUB 610 : IF
     PEEK (TRK ) = LT THEN 1050
1035 RESTORE
1040 TK = PEEK (TRK ) :ST = PEEK (SCT ) : GOTO
     1020
1050 HOME : PRINT "COPY^ DONE" : END
2000 DATA 255 ,255 ,255 ,255
2010 DATA 1^ CHANGES ,0 ,5 ,150 ,239
```

SOLO7
Fairfield, CT

## Lucky's Magic Hat

Here is a little softkey for Lucky's Magic Hat from Advanced Ideas.

1) Boot a normal DOS 3.3 disk.

2) Defeat DOS's error checking and return to basic with the following commands:

**CALL -151**
**B942:18**
**3D0G**

3) Run COPYA and copy both sides of the protected disk.

**RUN COPYA**

4) Use a sector editor (I use Copy ][ Plus) and search the disk for the following bytes.

**C6 2A D0**

5) I found them on track 09, sector 00. Change them to:

**6C 3A 00**

6) Write the sector back out to the COPYA disk and that's all there is to it! Side one is the only one protected.

**Notes:** I found after disassembling the code that the 4C 86 02 just after the code C6 2A D0 was the branch if not equal to (BNE) and sent the program into an endless loop. The code immediately after the BNE was the jump to the main program and just changing the BNE to the JMP was all that was needed to deprotect Lucky's Magic Hat.

The Pretzel
Bloomington, IL

## Conan APT's

Here are two APT's for Conan. When you get to level six, go down the first ladder and go as far to the right as possible, then turn to the left. This allows you to kill the eyes without getting killed. If you have fewer than 11 axes when you finish this level, stand on the spot where the axes appear and wait until you have 10 axes. Then throw an axe and catch it until you get 11. It helps out once you get to level 7!

Steven Nygard
Alberta, Canada

# input

## About Centipede

Even though it was not a runaway best seller, Centipede by Atari is a great favorite with children. The program can be nibble copied easily; but many people have one disk drive, and swapping disks repeatedly is a P-A-I-N. Because Centipede is a single load program, it can be easily copied with Wildcard II or using the methods explained in Dave Templin's extremely good article "Breaking In" in COMPUTIST No. 37. I chose, however, to take what was for me the easy way out.

Centipede uses changed address and data trailers: AA DE AB for both. Using the "Controller Writer" from COMPUTIST No. 16, I used Super IOB 1.5 to normalize the disk.

Also, Atari uses a slightly altered DOS. A routine on track $00, sector $00 checks for the altered marks and refuses to boot if it does not find them. Elsewhere certain disk commands are changed, BRUN, for example, is changed to BN. Not to worry. We do not want Atari's DOS anyway. ProntoDOS loads the game in about seven seconds as opposed to Atari's nineteen. The Hello program that BRUNs Centipede will, however, have to be changed.

Here, in cookbook fashion, is the method I used.

1) Initialize a blank disk (preferably with a fast DOS) and delete the Hello program.

```
INIT HELLO
DELETE HELLO
```

2) Install the Centipede controller into Super IOB 1.5 and let it work.

3) When Super IOB 1.5 has finished, load Centipede's Hello program and LIST it to make sure you have it (it's only a one line program). Type in the corrected line given below and SAVE it.

```
LOAD HELLO
LIST
10 HOME: CLEAR: PRINT CHR$
(4); "BRUN CENTIPEDE"
SAVE HELLO
```

That's all there is to it! Adults like the game, too.

## controller

```
1000 REM CENTIPEDE CONTROLLER
1010 TK = 17: LT = 24: 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 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: A$= "ALL DONE": GOSUB 450: END
5000 DATA 213, 170, 150, 213, 170, 173, 170,
     222, 170, 222
```

Robert J. Winterberg, Jr
West Milford, NJ

## Ghostbusters, Sea Dragon APT

Here is a cheat for Ghostbusters to give you any amount of money you want to start out with in the beginning of the game (this procedure assumes you have the deprotected version of Ghostbusters):

1) After the game asks you to enter your name, press [RESET]. You should now be in the monitor.

2) Edit locations $704C and $7051 to equal the amount of money you want (the amount can't exceed $999,999). This is done by first splitting your figure into three sections.

For example, let's take $500,000. It should be split like this: 50:00:00. The first part (50) should be stored at $704C by typing in the address at the monitor's prompt, a colon, and then the number: 704C:50.

The second part (00) should be stored at $7051 using the same method: 7051:00.

Nothing is done with the third part. It will always be whatever the second part is because of the way the game is programmed.

3) Now type: 8CEG.

You should again be presented with the introduction and the prompt to enter your name. When asked if you have an account answer "NO" and it will go on to tell you that the bank will forward you $10,000. Don't mind this as you will still have whatever you changed it to once the game starts.

Along with Phillip Lai's APT for Sea Dragon (COMPUTIST No. 38) where he stated that pressing [J] will give you 9999 air units instead of 6000, that's only for joystick play. [K] will give 9999 units for keyboard play, and [P] will give you 9999 air units for joyport play.

That's all for now! Until next time...

Sam Ismail

## To End Ultima IV

First, you must be a complete avatar, have the bell, book, candle, skull, 3-part key, and all 8 members of your party. Now, boot up Ultima IV. (J)ourney onward and insert the Britannia disk. Once the game is loaded, and everything is like it is when you play, hit CTRL-RESET.

Insert your Underworld disk into the drive and type "BLOAD END,A$8800". You will then crash into the monitor. Type "C050" <RETURN> and then "C057". Then finally type $8800G. The disk will load some stuff and you will be at the ending of Ultima IV! Now all that needs to be done is answer the questions. Just answer the 13 questions correctly and you win! Watch your typing, for if you make one mistake, you get kicked out of the Abyss.

That's all. Just read the text and you will win Ultima IV. Not so hard, eh?

Greg Poulos
Novi, MI

## PMH Pegasus

I recently purchased PMH Pegasus, a Lucasfilm Game from Electronic Arts, $29.95, and after many happy hours of playing this fantastic hydrofoil simulation I decided to try to back it up. The first thing I did was review my back issues of COMPUTIST, and in COMPUTIST No. 24 page 10, I found the answer. After reading Steve and Rod Smith's article on Electronic Arts software, I tried their softkey for Adventure Construction Set and it worked!!

Copy both sides of the game with any good disk copy program (I used Disk Muncher) and then make the following sector edits on the front side only.

```
Track $01, Sector $0E, bytes $47-$4A:
        from: 20 F8 A0 B0 to: 18 60 80 70
Track $07, Sector $0F: byte $52 from $D7 to $74
        byte $56 from $F3 to $04
```

That's all there is to it! See, it does pay to have those back issues handy. Keep up the good work. Oops, radar indicates an incoming missile, deploy the chaff rockets and prepare to fire.

Kenneth W. Anderson
Plano, TX

# readers' softkey & copy exchange

*Jason Rosenwald's softkey for...*

## The Color Enhanced Print Shop

*Broderbund Software Inc.*
*17 Paul Drive*
*San Rafael, CA 94903*
*$49.95*

**Requirements:**
64K Apple II
COPYA
A sector editor
A disk searcher

The New Color Enhanced Version of the Print Shop is an exellent program. It can be used to make signs, greeting cards, and banners. It has many new enhancements that the old Print Shop did not have like the ability to save the whole picture to disk, previewing the picture before printing, and of course, with an Imagewriter II color printer or compatible, the ability to print in seven different colors. One setback is that the new version does not allow you make one backup copy like the old one did. You must send in the order form and wait a couple of weeks. But of course, after finishing this procedure you will no longer need the order form.

I started out by copying the disk with Locksmith fast copy to see where the protected tracks live. Just like the old Print Shop, track $22 was the only one. I then booted the disk and found it loaded the title page and checked for a nibble count and then just hung there. I used Track Finder from COMPUTIST No. 37 and discovered it checked for some bytes on track $22. I bit copied $22 but still had no luck.

After boot code tracing the program I found the start of the nibble count at $B619. It fills page $BB00 with $FFs, reads the disk for four markers ($D4,$D5,$DE,$D4) and reads a pair of 4+4 encoded bytes. It then reads two more markers ($F5,$AA) at $B6B5.

I then looked for byte sequence $A0 00 A9 FF, (LDY #$00,LDA #$FE) and put an RTS ($60) at $B619. I booted up the disk and it skipped the nibble count routine and worked perfectly. Another win in the deprotection wars.

### The Cookbook Method

1) Copy the original disk with COPYA or another whole disk copy program.

2) Get out your disk searcher and scan for the bytes $A0 00 A9 FF, and change the first byte ($A0) to a ($60). Mine was on track $01, sector $0F, byte $19.

3) Enjoy your freshly cracked Print Shop.

*Lyle Marentette's softkey for...*

## Video Vegas

*Baudville*
*1001 Medical Park Dr. S.E.*
*Grand Rapids, MI 49506*
*(616)957-3036*
*$29.95*

**Requirements:**
COPYA
A sector editor

Video Vegas is one of Baudville's new Hacker Jack series of programs. It is designed so that you can "get into" the programs (written in BASIC) and learn advanced programing techniques. "If you are a serious hacker, you can even customize the program by changing the graphics or program code", Baudville says. It's a great program, with fantastic graphics.

You can "stroll" through the casino, and play Lucky 7 (the slot machine), poker, keno, and, of course, blackjack (1 to 4 decks). While playing blackjack, you can hit the ? key and get a running card count. (a great way to learn how to count cards, and beat the casino).

The protection is almost the same as Take 1. It deals with three conditional branches. (see the Take 1 softkeys in COMPUTIST Nos. 25 and 33). To make an unprotected copy of Video Vegas follow these steps:

1) Make a copy of the original Video Vegas with COPYA or any copier.

2) Use your favorite sector editor, and make the following changes to the copy:

Track 0, sector $6:

| Byte# | From: | To: |
|---|---|---|
| $11 | $D0 | $EA |
| $12 | $F7 | $EA |
| $2E | $D0 | $EA |
| $2F | $D5 | $EA |

Track 0, sector $F:

| Byte# | From: | To: |
|---|---|---|
| $08 | $D0 | $EA |
| $09 | $F7 | $EA |

3) Enjoy!

# readers' softkey & copy exchange

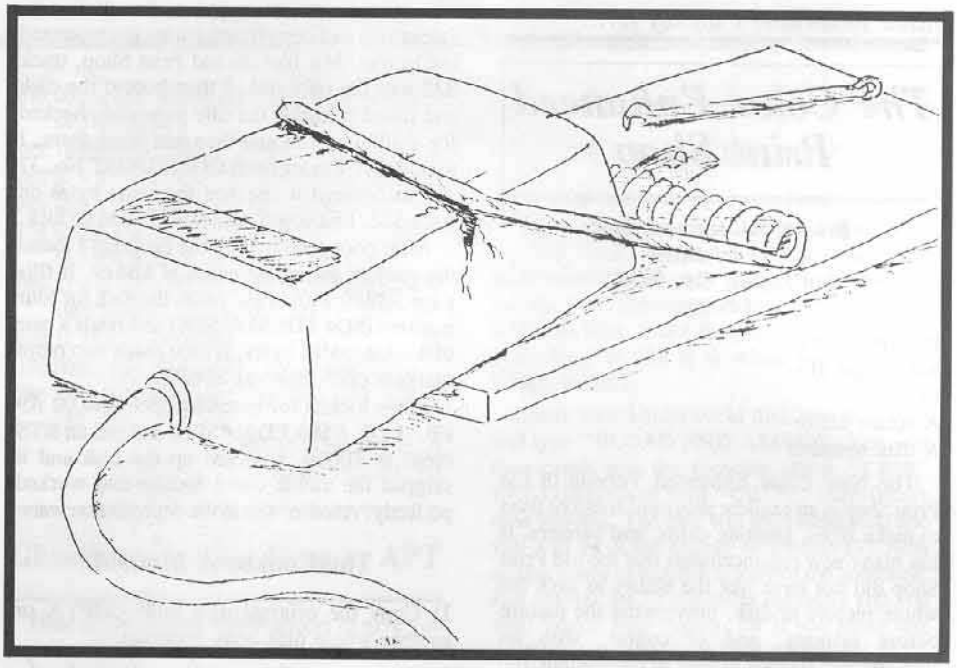
*Jason Lee's softkey for...*

## *The Handlers*

**Word Handler v4.4**
**List Handler v1.4**
**Spell Handler v3.0**

*Advanced Logic Systems*
*1195 East Arques Avenue*
*Sunnyvale, CA 94086*

**Requirements:**
Apple II computer
COPYA or equivalent
A sector editor
Three double-sided disks



The Handler set is one of the favorites in word processors in our school system. It is even used by the disabled learning students in the community. Thus, it was very important that the Dept. of Education maintain backups in any case of disk failure.

Before everyone gets all exited let me forewarn you that previous versions of The Handlers by SVS cannot be copied using the following method. A couple years ago, ownership switched from Silicon Valley Systems to Advanced Logic Systems. Neither versions are easy to duplicate with a bit copier.

First attempts at copying were plain duplication with all sorts of bit copiers and sector copiers. Some of them worked, but they were extrememly tedious. So, disregarding the message about copy protection in the manual, I was left with one alternative: crack them.

**Spell Handler v3.0** does not seem to be protected at all and will copy with any COPYA equivalent.

For **Word Handler**, by simply following the execution (it isn't all that hard since the disk is readable by normal DOS) I came to the first bit of code from the file WH:

```
46FD-  JSR $5500
4700-  BVC $473B
4702-  INC $03F4
4705-  INC $0427
4708-  JMP $4705
```

Well, any novice at machine language can figure out that if the branch isn't taken, the reset vector is altered and the program jumps into an endless loop. During your first try at backing this up you may have noticed a character in the upper right hand corner of the text screen alternating quickly. The only way out of it is RESET, which will cold start the system. So, I felt it necessary to branch regardless of the overflow flag.

1) Copy both sides of Word Handler onto your double sided disk using something like COPYA or Locksmith 6.0 Fast Disk Backup.

2) The following patch will change the BVC (branch on overflow clear- $50) to BVS (branch on overflow set- $70). It can be done using eithor of two methods:

**a-** do the first sector edit on the 66 and 40-column side and the second sector edit on the 80-column format side of your duplicate-

| Side | Trk | Sec | Byte | From | To |
|---|---|---|---|---|---|
| 66/40 | $17 | E | $0A | 50 | 70 |
| 80col | $17 | E | $07 | 50 | 70 |

**b-** or boot up the 66 and 40-column side and press 4 to quit. Load the file WH and make the patch and then re-save it. Also do the same for the 80-column side.

```
BLOAD WH
CALL -151
4703: 70
BSAVE WH,A$46FD,L$70C
```

(flip disk)

```
BLOAD WH
4700: 70
BSAVE WH,A$46FD,L$70C
```

For **List Handler**, the protection is very similar to that of Word Handler. The UTILITIES side is not protected so copy that onto the back side of a disk. For the front side you may have noticed the same text screen garbage if you simply tried to COPYA it. Just looking at the machine language code of the only substantial program on the disk I came to this:

```
3000-  INC  $03F4
3003-  LDX  #$FF
3005-  TXS
3006-  JSR  $3100
3009-  BVC  $3011
300B-  INC  $0427
300E-  JMP  $300B
3011-  JMP  $3800
```

Again, here is the same loop at $300B. The way to counter that is to change the BVC (op code $50) to a BVS (op code $70).

1) Copy both sides of List Handler using any disk copier.

2) Exit into DOS and load the program MAIN3 on the first side.

```
BLOAD MAIN3
```

3) Go into the monitor and make the necessary changes. Then resave to program back onto the disk.

```
CALL -151
3009:70
BSAVE MAIN3,A$3000,L$5D00
```

That's about all. Everything seems to work fine.

# readers' softkey & copy exchange

*Jim S. Hart's softkey for...*

## K.C.'s Deals On Wheels v1.0

*K.C.'s Deals on Wheels-*
*A Computerized Accounting Simulation*

*Richard D. Irwin, Inc.*
*Homewood, IL*

**Requirements:**
K.C.'s Deals on Wheels original disk
COPYA or similar
A sector editor with search capability

K.C.'s Deals on Wheels is an accounting simulation used to help students learn a little of what it is like to own your own business. You must keep track of financial statements, accounts receivable & payable, and several other items. The program is an excellent one. It comes with a book that contains exercises for students to follow. The only bad thing is that when and if the disk crashes, you must go back to the book store and buy another book & disk for around $25. I have seen several crashes and in each case the student dropped the course because he/she did not want to shell out another $25. It's sad when things like this happen because you can't make a backup of your disk.

The three bit copiers that I have access to, Copy II Plus, Locksmith 6.0, and EDD 3, could not copy the disk. EDD 3 came the closest but the copy still wouldn't work. On a hunch I tried to copy the disk with COPYA and it copied with no problems! Things couldn't be this easy, and when the copy was booted, the message 'ILLEGAL COPY' came up with a compliment of beeps to go along with it. Ah hah! A nibble count is probably being executed somewhere and not being satisfied.

Apple PASCAL, which is the disk's format, accesses the disk drive using direct addressing, i.e. to turn on the drive you would look for the byte sequence E9 C0 instead of 89 C0. I found some interesting code on track $0F, sector $09 and track $10, sector $06. The way I disable nibble counts is to modify the code so that no matter what happens, a 'correct' result will occur. I usually do not jump over the code since my way is more ironic. Playing around with the nibble count code here proved to be frustrating so jumping over the code became a neccessity. The bytes 2C E9 C0 turn on the drive. The bytes 2C E8 C0 turn off the drive. What I did was to replace the 2C E9 C0 with a clear carry (18) and then a branch on carry clear (90 5A) so that it would jump to some utility code right before the code that turned off the drive (2C E8 C0). I did this to both nibble counts and voila', the copy works! All you have to do is to search for the byte sequence 2C E9 C0 on the copied disk and change it to 18 90 5A. On the instructor's disk there is a similar check but it only occurs once on the disk at track $0F, sector $02. If you want to see what a nibble count looks like, disassemble the code after the changes you made. This might help in finding other nibble counts down the road. Enjoy your backup and the peace of mind that goes with it.

### Step By Step

1) Copy the original with any whole disk copier such as COPYA.

2) Make the following sector edits to the copy:

| Trk | Sector | Bytes | From | To |
|---|---|---|---|---|
| $0F | $09 | $64-66 | 2C E9 C0 | 18 90 5A |
| $10 | $06 | $38-3A | 2C E9 C0 | 18 90 5A |

3) If you have the instructor's disk, follow step #1 and then make this sector edit to the copy:

| Trk | Sector | Bytes | From | To |
|---|---|---|---|---|
| $0F | $02 | $16-18 | 2C E9 C0 | 18 90 5A |

*The Guildmaster's softkey for...*

## Law of the West

*Accolade*
*20863 Stevens Creek Blvd.*
*Cupertino, CA 95014*

**Requirements:**
COPYA
A sector editor
Law of the West disk

After examining my newest disk, Law of the West, I discovered it was written in standard DOS 3.3 (16-sector) format. I naturally tried to back it up with a simple copy program like COPYA. However, this method failed, so after an hour or so of disk searching, I developed the following softkey:

1) Copy the disk with COPYA.

2) Perform the following sector edits to disable the copy protection:

| Track | Sector | Byte | To |
|---|---|---|---|
| 0 | B | $22 | $18 |
| | | $23 | $60 |

3) Write the changes back to the disk.

4) Enjoy your deprotected disk.

Note: Because of a highly modified, if not completely re-written DOS, Law of the West is quite difficult to effectively modify.

# readers' softkey & copy exchange

*Jim S. Hart's softkey for...*

## *Break the Bank Blackjack*

*Gentry Software*
*9411 Winnetka Avenue*
*Chatsworth, CA 91311*
*$29.95*

**What's needed:**
Break the Bank Blackjack original
A blank disk
Super IOB w/Swap controller
A way into the monitor or the ability to open the disk drive door
A fast DOS (optional)

Break the Bank Blackjack (henceforth known as BBB) is a blackjack simulation put out by the folks at Gentry Software. I cannot really call it a game since the primary objective of it is to teach the user how to 'count cards'. For the uninitiated among us, this method of play allows one to have a better winning percentage against the house which translates to more money. Like most other programs however, it is protected but has that stopped us COMPUTIST loyalists before? NO. That is what this magazine is for: to give you piece of mind with your original user-FIENDLY (to borrow a phrase) software. Follow the steps below to convert it to user-FRIENDLY software.

1) Initialize the blank disk, preferably with a fast DOS, and delete the hello program:

```
INIT HELLO
DELETE HELLO
```

2) Now it is time to capture the BBB's RWTS. There are several ways to do this outlined in prior COMPUTIST articles. I have three favorites: boot tracing, RESET into the monitor, and what I call 'open the disk drive door when the Applesoft prompt appears'. Instead of getting complicated with all that boot tracing stuff, let's do it the easy way. Boot your original BBS disk and when the Applesoft prompt appears, open the drive door. After a few seconds of rattling the computer will beep and give you an I/O error (those of you with a way into the monitor at will should RESET into it when the title screen comes up).

3) Now we have to move the RWTS to a safe place so that a subsequent boot will not wipe it out:

```
CALL -151
1900<B800.BFFFM
```

4) Insert your Super IOB disk into the drive and boot it:

```
C600G
```

5) Save the RWTS to disk:

```
BSAVE RWTS.BBB,A$1900,L$800
```

6) What we have to do now is load up Super IOB and install the SWAP controller into it. This can be done by LOADing two separate files or using a textfile to merge them or any one of a number of other ways. After you have loaded them, modify line 1010 so the variable TK is equal to three (i.e. TK=3) and copy the rest of the line. Now add this line to load in the RWTS:

```
10010 PRINT CHR$( 4); "BLOAD
RWTS.BBB,A$1900'
```

7) Copy the disk:

```
RUN
```

8) When Super IOB is finished, you will be the proud owner of an unprotected disk. Put the original away in a safe place and use the backup with confidence!

*Jim S. Hart's softkey for...*

## *Foundation Course In Spanish, 6th Edition*

*DCH Educational Software*
*D.C. Heath & Company*

**Requirements:**
4 Foundation Course In Spanish original double-sided disks
8 blank disk sides (4 notched blank disks)
Super IOB 1.5

The Foundation Course in Spanish series is designed to help students learn to use the Spanish language interactively with the computer. It is a well designed set but alas, the disks are not COPYAable. This is a no-no in a classroom environment so it is time to get rid of the protection. Inspection of the disks with a nibble editor shows altered epilogs and checksums. (For an explanation of what epilogs and checksums are, I refer you to other COMPUTIST articles and the book *Beneath Apple DOS* by Worth & Lechner.) It is a simple matter to convert these types of disks to normal using a slightly modified Fast controller & Super IOB. In the controller listed below, the POKE in line 1020 disables error checking for the reading of the protected disk and the one in line 1030 enables error checking for writing to the copy. To convert the originals to unprotected (COPYAable) format, install the controller below into Super IOB v1.5 and use it on each one of the eight sides. When you are done you will be able to let the students use the copies while the originals can be safely hidden away from prying fingers. Enjoy!

### Step By Step

1) Install the controller below into Super IOB v1.5 and RUN it.

2) Copy each of the eight sides onto the blank sides, using the Format option to format the blank if necessary.

3) Below is a list of the sector edits made by the controller for your viewing pleasure. That's it!

| Track | Sector | Byte | From | To |
|---|---|---|---|---|
| $00 | $0E | $EB | $AA | $DE |
| $00 | $0E | $F5 | $DE | $AA |
| $01 | $0E | $CE | $AA | $DE |

### controller

```
1000 REM FOUNDATION COURSE IN  SPANISH
1010 TK = 0 :LT = 35 :ST = 15 :LS = 15 :CD = WR
     :FAST = 1
1020 POKE 47426 ,24 : GOSUB 490 : GOSUB 610
1025 T1 = TK :TK = PEEK (TRK ) - 1 : RESTORE :
     GOSUB 310 :TK = T1
1030 POKE 47426 ,56 : GOSUB 490 : GOSUB 610 :
     IF PEEK (TRK ) = LT THEN 1050
1040 TK = PEEK (TRK ) :ST = PEEK (SCT ) : GOTO
     1020
1050 HOME : PRINT "COPYDONE." : END
5000 DATA 3^ CHANGES
5010 DATA 0 ,14 ,235 ,222
5020 DATA 0 ,14 ,245 ,170
5030 DATA 1 ,14 ,206 ,222
```

### controller checksums

```
1000 - $356B      1050 - $E685
1010 - $2544      5000 - $B63E
1020 - $B5FB      5010 - $819E
1025 - $BE24      5020 - $F79E
1030 - $C238      5030 - $6DBA
1040 - $B1CA
```

# readers' softkey & copy exchange

*Charles Taylor's softkey for...*

## *OGRE*

*Origin Systems*
*340 Harvey Road*
*Manchester, NH 03103*

**Requirements:**
64K Apple II
A blank disk
Super IOB 1.5

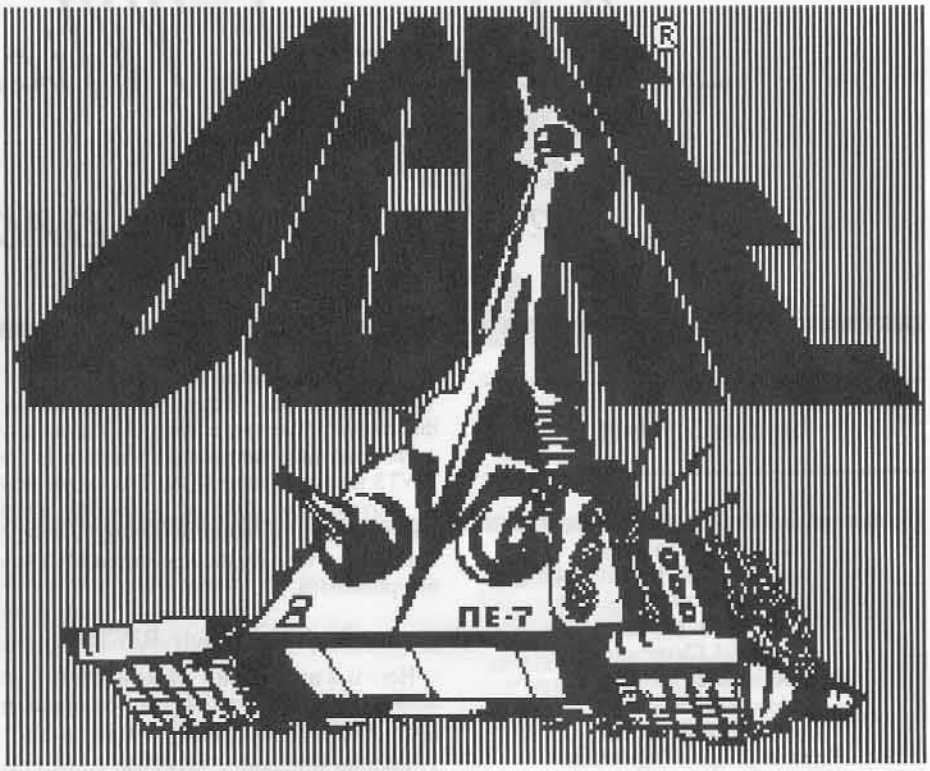
OGRE is published by Origin Systems, distributed by Electronic Arts, and looks like a Strategic Simulations board game. Guess whose copy protection they used? (Answer below.) It is a tank battle game on hex "squares" that can be a two player game or one player game against a computer OGRE. The OGRE is a super robot tank that can barely be destroyed by atomic weapons.

The answer to the riddle is "none of the above". It is copy-protected in a manner similar to a variety of other games (such as Penguin Software's games), using alternating address prologues of D5 AA 96 on even numbered tracks and D4 AA 96 on odd numbered tracks. The address epilogues vary from sector to sector, but the data marks are normal. The sync bytes in the large gap are 84 instead of FF, which is probably the reason that bit copiers with default parms can't copy it. Not wanting to add to my IOB controller collection, I searched my back issues of COMPUTIST until I came up with Mr. Strelchun's Dragonworld controller from issue #30, page 28. It worked without any further sector edits, nibble count searches, or patches.

Presented here is a faster version of that controller. It is just the Fast controller with an additional line (1015).

### controller

```
1000 REM OGRE CONTROLLER
1010 TK = 0 :LT = 35 :ST = 15 :LS = 15 :CD = WR
     :FAST = 1
1015 POKE 47507 ,0 : POKE 47517 ,0 : POKE
     47444 ,41 : POKE 47445 ,42
1020 GOSUB 490 : GOSUB 610
1030 GOSUB 490 : GOSUB 610 : IF PEEK (TRK ) =
     LT THEN 1050
1040 TK = PEEK (TRK ) :ST = PEEK (SCT ) : GOTO
     1020
1050 HOME : PRINT "COPYDONE" : END
```




### controller checksums

```
1000  - $356B     1030  - $BB5F
1010  - $2544     1040  - $F2E9
1015  - $0B4B     1050  - $D6F4
1020  - $BB45
```

*Paul Giguere's softkey for...*

## *Puzzles and Posters*

*MECC*
*(Minnesota Educational Computing Consortium)*

**Requirements:**
Super IOB 1.5
A blank disk

All that this program has for protection is an altered address marker.

1) Initialize a disk with a HELLO program, then DELETE it.

2) Install the controller below into Super IOB and RUN it. That's it, another broken program!

### controller

```
1000 REM MECC CONTROLLER
1010 TK = 3 :LT = 35 :CD = WR :MB = 151 : ONERR
     GOTO 550
1020 ST = 0 :T1 = TK : GOSUB 490 : RESTORE :
     GOSUB 190 : GOSUB 210 : GOSUB 170
1030 GOSUB 430 : GOSUB 100 :ST = ST + 1 : IF ST
     < 16 THEN 1030
1040 IF BF THEN 1060
1050 ST = 0 :TK = TK + 1 : IF TK < LT THEN 1030
1060 GOSUB 230 :TK = T1 :ST = 0 : GOSUB 490
1070 GOSUB 430 : GOSUB 100 :ST = ST + 1 : IF ST
     < 16 THEN 1070
1080 ST = 0 :TK = TK + 1 : IF BF = 0 AND TK < LT
     THEN 1070
1090 IF TK < LT THEN 1020
1100 HOME :A$ = "COPY^ DONE" : GOSUB 450 :
     END
5000 DATA 170 ,213 ,150 ,213 ,170 ,173 ,222
     ,170 ,222 ,170
```

### controller checksums

```
1000  - $356B     1060  - $6AE6
1010  - $5E3F     1070  - $22FD
1020  - $B92C     1080  - $54D8
1030  - $E2AA     1090  - $7FC2
1040  - $2463     1100  - $D9B0
1050  - $E2BC     5000  - $0FF5
```

# Advanced Microsytems Technology programs

by Jim S. Hart

**Fundamentals Of:**
**AC Circuit Analysis**
**Basic Electricity**
**Direct Current**
**Electronic Devices**
**Industrial Electronics**
**Microcomputer Interfacing**
**Microprocessors**
**Operational Amplifiers**
**Pulse & Logic Waveforms**
**Regulated Power Supplies**

*Advanced Microsystems Technology*
*Box 1942*
*Bowling Green, KY*

**Requirements:**
Ability to Reset into the monitor or boot code trace
Advanced Microsystems Technology original program disk(s)
Super IOB 1.5
A blank disk for each program
A fast DOS (not necessary but helpful)

Advanced Microsystems Technology puts out some excellent educational programs in the field of electrical concepts. The programs themselves teach students about electrical concepts, microprocessors, and current among other topics. An instructor can even keep track of how well a student did via a password to get to a 'scorekeeping' program. My only complaint is that the programs are supplied on copy-protected disks. The company has a good policy in regards to receiving backups from them in case your original bites the dust. This takes time though, so I decided to go about deprotecting the disks.

## What To Look For

Here are some short things to try first:
► Can the disk be CATALOGed?
► Does COPYA copy the disk?
When you try both of these you will discover that neither of them are successful. This usually means that there are format alterations in the RWTS section of their DOS. The easiest way to convert these disks is to use the Swap controller & Super IOB. To do this, however, requires that we have their perverted RWTS in our possession.

## Grabbing Their RWTS

Here is a method I use to get a RWTS into memory. It involves a little boot code tracing (just enough to get your feet wet).

1) Boot up with normal DOS and go into the monitor:

**CALL -151**

2) Move the disk controller card ROM into main RAM so that it can be modified to suit our needs:

**9600<C600.C6FFM**

3) Fix this code so that instead of booting the disk it returns control to us:

**96FA:98 N 9801:AD E8 C0 4C 59 FF**

4) Put the original (write protected, of course) into the drive and boot it:

**9600G**

5) The drive will turn on for a second, rattle, beep and go into the monitor. First move the BOOT1 code (which loads in the RWTS) to a safe spot. Next, we need to modify BOOT1 so it will load in the RWTS and then go into the monitor again:

**9800<800.8FFM**
**980E:90**
**984A:4C 00 93**
**9300:AD E8 C0 4C 59 FF**

6) Execute the modified boot one more time:

**9600G**

7) When the computer beeps you will have the RWTS in memory at $B800-BFFF. All that needs to be done is to move it to a safe place so that a subsequent boot will not erase it:

**1900<B800.BFFFM**

8) Boot a normal DOS disk and save the RWTS:

**C600G**
**BSAVE RWTS.PROG,A$1900,L$800**

## Copying The Originals

1) Install the Swap or New Swap controller into Super IOB.

2) Add these two lines to the merged program and then copy the original disk:

```
1015 TK = 3
10010 PRINT CHR$ (4 ); "BLOAD^
    RWTS.PROG,A$1900"
```

RUN

## Alternate Method

If for some reason the above method does not work, here is a short alternate method that should work. (NOTE: You'll be surprised at how many educational disks you can deprotect by this alternate method)

1) Boot a (preferably fast) DOS 3.3 disk.

2) Load up Super IOB, add the following controller to it, execute it, and be sure to use the Format option to format the copy disk:

```
1000 REM COPY DISK W/NO ERROR CHECKING & NO
    DOS
1010 TK = 3 :LT = 35 :LS = 15 :ST = 15 :FAST = 1
1020 POKE 47426 ,24 : GOSUB 490 : GOSUB 610
1030 POKE 47426 ,56 : GOSUB 490 : GOSUB 610 :
    IF PEEK (TRK ) = LT THEN 1050
1040 TK = PEEK (TRK ) :ST = PEEK (SCT ) : GOTO
    1020
1050 HOME : PRINT "COPYDONE." : END
```

The copied disk is now deprotected. This procedure works for all (to my knowledge) disks by Advanced Microsystems. I hope the method presented for capturing a RWTS proves helpful to the readers out there.

# revisited softkey for...

# Word Attack

by Jim S. Hart

*Davidson & Associates*
*3135 Kashiwa Street*
*Torrance, CA 90505*
*$49.95*

**Requirements:**
Word Attack! original that wouldn't work with softkey in COMPUTIST No. 28
Super IOB 1.5
Two blank disk sides
COPYA
Bag Of Tricks from Quality Software
A fast DOS (optional) like Pronto-DOS or Diversi-DOS

Word Attack! is a program that "teaches youngsters new words, meanings, and their meanings in an interesting and exciting way" (to paraphrase from the packaging). It is an excellent program. It needed to be deprotected, however, since children would be using it. I scanned through my old issues of COMPUTIST and in issue No. 28 I came across Dave Stanton's softkey. It was well done but I could not get it to work no matter how hard I tried. Checking the original out with a nibble editor confirmed my suspicions: a different protection scheme. The address prologues change on every track (after track $03), the data prologue had been changed to D5 AA B5, and the epilogs and checksums were a mess. Yuch. It seemed that I would be spending some time with this disk writing down all of the changing address prologues and the like when I noticed something peculiar. The address prologues on the first couple of tracks were identical to the ones on Ultima IV. For the heck of it, I tried using the Ultima IV controller on the Word Attack! original and guess what? It worked like a champ! A fast DOS was added to the copy and then it was booted. Things looked fine, but after a little while the disk rebooted itself. Oops, missed something. Reviewing the original with the nibble editor showed the disk's volume number to be a zero. This is normally an illegal volume number but luckily I have access to Quality Software's Bag Of Tricks. After changing the volume number to zero and making a few changes to the BASIC programs so it wouldn't cause a reboot, the disk booted fine and the programs ran without a hitch. A bit of advice: get as many back issues as you can and read up on the different protrection schemes used. It may come in handy later on.

## Step by Step

1) Boot up DOS (preferably a fast one) and initialize the front side of the blank (don't forget the [CTRL-Z] in the filename!):

**INIT H[CTRL-Z]ELLO**

2) Load up Super IOB 1.5 and install the controller following this article (it's the Ultima IV controller from COMPUTIST No. 28).

3) Use Super IOB to copy the front side onto the blank.

4) Use COPYA to copy the back of the original onto the other blank side.

5) Boot up the Bag Of Tricks and choose the INIT program. Change the volume number to zero and "re-volume" the front side of the copied disk.

6) Load up the BASIC file W[CTRL-Z]ORD ATTACK!, eliminate the reboot routine and protection checks, and save the modified file back to the disk.

**LOAD W[CTRL-Z]ORD ATTACK!**
**127**
**303 RETURN**
**304 REM**
**305 REM**
**306 REM**
**SAVE W[CTRL-Z]ORD ATTACK!**

7) Just for fun, don't make the changes to the Word Attack! file and see how far you get. Line numbers 127 and 303 check for the original operating system while line numbers 304 to 306 are the reboot routine.

8) You're done! Hide the original and enjoy your liberated backup.

## controller

```
1000 REM WORD ATTACK (ULTIMA IV)
1010 TK = 3 :LT = 4 :ST = 15 :LS = 15 :CD = WR
1020 POKE 47405 ,24 : POKE 47406 ,96 : POKE
     47497 ,24 : POKE 47498 ,96
1030 POKE 47829 ,3 :T1 = TK : GOSUB 490 :
     GOSUB 210
1040 GOSUB 190 : GOSUB 610
1050 TK = TK + 1 :LT = LT + 1 : IF PEEK (BUF ) <
     MB AND TK < 35 THEN 1040
1060 POKE 47405 ,208 : POKE 47406 ,19 : POKE
     47497 ,208 : POKE 47498 ,183 : POKE 47829
     ,213 : GOSUB 230
1070 TK = T1 :LT = 35 : GOSUB 490 : GOSUB 610 : IF
     PEEK (TRK ) = LT THEN 1090
1080 TK = PEEK (TRK ) :ST = PEEK (SCT ) :LT = TK
     + 1 : GOTO 1020
1090 HOME : PRINT "COPY^ DONE,^ DOS^ NOT^
     COPIED." : END
5000 DATA 213 ,170 ,181 ,215 ,170 ,151 ,213 ,170
     ,150 ,213 ,170 ,151 ,215 ,170 ,150
5010 DATA 215 ,170 ,151 ,221 ,170 ,158 ,221 ,170
     ,159 ,213 ,170 ,181 ,223 ,170 ,158
5020 DATA 223 ,170 ,159 ,221 ,170 ,158 ,221 ,170
     ,159 ,223 ,170 ,158 ,223 ,170 ,159
5030 DATA 213 ,170 ,150 ,213 ,170 ,181 ,213 ,170
     ,151 ,215 ,170 ,150 ,215 ,170 ,151
5040 DATA 213 ,170 ,150 ,213 ,170 ,151 ,215 ,170
     ,150 ,215 ,170 ,151 ,213 ,170 ,181
5050 DATA 221 ,170 ,158 ,221 ,170 ,159 ,223 ,170
     ,158 ,223 ,170 ,159 ,221 ,170 ,158
5060 DATA 221 ,170 ,159 ,223 ,170 ,158 ,213 ,170
     ,181 ,223 ,170 ,159 ,245 ,170 ,182
5070 DATA 245 ,170 ,183 ,247 ,170 ,182
```

## controller checksums

```
1000 - $356B      1090 - $C930
1010 - $3189      5000 - $FEC0
1020 - $C562      5010 - $4560
1030 - $545E      5020 - $9B9E
1040 - $DDB4      5030 - $0B30
1050 - $A5C8      5040 - $20BC
1060 - $044B      5050 - $165F
1070 - $B732      5060 - $20B9
1080 - $045C      5070 - $FA72
```

# The Shift Key and the Lowercase Option for the ][ Plus

by Gary Kowalski

**Requirements:**
Rev 7 or later Apple II or II+
Soldering iron
Double pole, double throw (DPDT) switch
Access to an EPROM programmer and 2716 type EPROMs,
or RAMcard and lowercase chip

***Warning:*** *The procedure described in this article should only be attempted by those persons having computer hardware experience. Softkey Publishing assumes no liability for damage done to the computer if this procedure is followed.*

One of the most annoying things about the old Apple IIs is the lack of proper shift key operation with respect to lowercase character generation. If you want lowercase you either buy an 80 column card or install a lowercase chip and then you wire the famous "shift key modification" that wires the shift key to game controller #3 pushbutton input. However, you soon find out that word processors are the only programs that recognize the shift key modification, so you can't type lowercase from the keyboard for BASIC or other applications.

About the same time Apple introduced the Rev 7 version of the Apple II motherboard, they also introduced the two piece keyboard. To see if you have one of these, pop off the top of your Apple and look at the underside of the keyboard. If you see a 25 pin comb connector that holds a small circuit board to the underside of the keyboard, then you can get true shift key and lowercase for your machine.

Apple spotted the lowercase shortcoming and introduced a lowercase option with the two piece keyboard but they failed to tell anyone about it. There is nothing about it in my Apple II Reference Manual or addendum to the manual even though I should have the latest revision of this manual because I bought my computer just before the IIe reached the market. I learned about this lowercase option from "The Apple II Circuit Description" by Winston Gayler, Howard W. Sams & Co.

To get your normal shift operation, remove the screws that hold the plastic case of the computer to the metal baseplate. Before separating the case from the baseplate, carefully remove the ribbon cable that connects the keyboard to the motherboard. Next remove the small circuit board from the keyboard by squeezing the plastic mounts and pulling gently to separate them. On the circuit board are two "bowties" that must be cut (see figure 1).

Next, mount a DPDT switch on this circuit board where Apple marked the board for it (see figure 1) or if you prefer, mount it on the case of your Apple near the keyboard. This will be a CAPS LOCK switch and when in the CAPS LOCK postion, the cut bowties will be "re-connected". In the other switch position, the keyboard will generate lowercase unless the shift key is pressed. See figure 2 for the switch connections to the circuit board. After doing this, the keyboard will generate the lowercase but you will not see it for two reasons:

1) You must have lowercase in the character generator ROM.

2) The F8 ROM converts lowercase to uppercase.

The first problem can be taken care of by buying a lowercase chip (or programming a 2716 EPROM) and replacing the 2316 character generator ROM at motherboard position A5 (directly under the keyboard) with it. I consider the second reason to be an unexcusable rudeness on Apple's part. If the early Apples could not generate lowercase from the keyboard, why must the KEYIN routines filter them out?

To stop the F8 from converting lowercase to uppercase you can use your language card to change the monitor KEYIN routines or program a new F8 EPROM. The minimum change to make is byte $FD83 from $DF to $FF:

```
Change:   AND $DF   ;Convert to uppercase
   To:    AND $FF   ;Don't convert
```

**FD83: FF**

One remaining problem occurs when the cursor is on a lowercase character. The cursor will appear as a flashing "special" character (!,',#,$,etc) because the lowercase character ROMs have inverse special characters and not inverse lowercase. You can either ignore this or re-write the KEYIN routines (COMPUTIST #19 has some good routines to consider).

To make the modifications using your RAM card, do the following:
(enter the monitor)

**CALL -151**

(write enable RAM card)

**C081**
**C081**

(copy monitor routines to Ramcard)

**F800<F800.FFFFM**

Now make the changes to the desired routines. You won't be able to see your changes in operation or list them until you disable the ROM and enable the RAM card for reading:

**C083**

Note that the changes made via RAM card disappear upon power off so you have to re-load your changes into the RAM card each time you power up.

The accompanying hexdump provides the necessary changes to a normal character generator ROM to make it a lowercase character ROM. The listing assumes that the ROM image starts at $2000.

## Lowercase Character Modification

```
2700: 80 80 80 80 80 80 80 80   $8097
2708: 80 80 80 98 84 9C A4 9A   $82D4
2710: 80 A0 A0 BC A2 A2 A2 BC   $6AC3
2718: 80 80 80 9C A2 A0 A0 9E   $9D12
2720: 80 82 82 9E A2 A2 A2 9E   $A472
2728: 80 80 80 9C A2 BE A0 9C   $C9C1
2730: 80 8C 92 90 B8 90 90 90   $84BD
2738: 80 82 9C A2 A2 9E 82 9C   $8C52
2740: 80 A0 A0 AC B2 A2 A2 A2   $A242
2748: 80 88 80 88 88 88 88 88   $E2CA

2750: 80 84 80 84 84 84 A4 98   $7E50
2758: 80 A0 A0 A4 A8 B0 A8 A4   $CEF8
2760: 80 98 88 88 88 88 88 9C   $1E5A
2768: 80 80 80 BC AA AA AA AA   $502A
2770: 80 80 80 9C 92 92 92 92   $46A2
2778: 80 80 80 9C A2 A2 A2 9C   $FE25
2780: 80 80 80 BC A2 BC A0 A0   $276D
2788: 80 80 80 9E A2 9E 82 82   $D306
2790: 80 80 80 AC B2 A0 A0 A0   $D2E0
2798: 80 80 80 9C A0 9C 82 9C   $978C

27A0: 80 80 90 BC 90 90 90 8C   $9F24
27A8: 80 80 80 A4 A4 A4 A4 9C   $677E
27B0: 80 80 80 A2 A2 A2 94 88   $3216
27B8: 80 80 80 A2 A2 AA AA 94   $285A
27C0: 80 80 80 A2 94 88 94 A2   $087E
27C8: 80 80 80 A2 A2 94 88 B0   $8BE1
27D0: 80 80 80 BC 84 98 A0 BC   $91A5
27D8: 80 86 88 84 98 84 88 86   $3F05
27E0: 80 88 88 88 80 88 88 88   $7F15
27E8: 80 B0 88 90 8C 90 88 B0   $8135

27F0: 80 90 AA 84 80 80 80 80   $108D
27F8: 80 AA 94 AA 94 AA 94 AA   $924D
```

**Related articles:** "Towards a Better F8 ROM" in COMPUTIST No. 19 shows some additional changes to your autostart F8 ROM. In the same issue, "Double Your ROM Space" tells you how to add a 2716 or 2732 EPROM to your Apple II (not //e or //c). "More ROM Running", COMPUTIST No. 34, details the use of the language card and how you can use it to simulate a modified ROM.

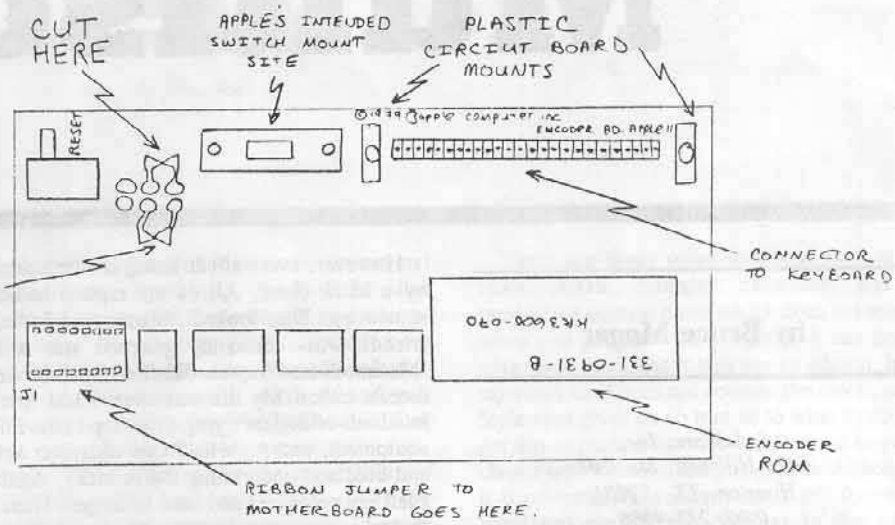



FIGURE 1

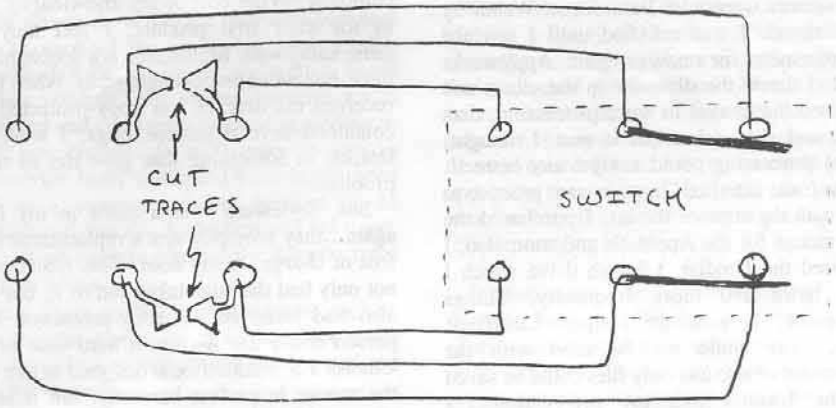



FIGURE 2 CAPSLOCK SWITCH IN LOWERCASE POSITION

# Multiscribe:

by Bruce Mager

*StyleWare, Inc.*
*6427 Hillcroft, Ste. 201*
*Houston, TX 77081*
*(800) 233-4088*
*In TX (713) 771-4627*
*$59.95*

**Hardware Requirements:**
Apple //c or enhanced //e
Apple Mouse (optional)
Printer (not required, but highly recommended)
A truckload of imagination

It was quite a while ago that I was acquainted with Multiscibe. My first word processing experiences were with Bank Street Writer by Broderbund. I was satisfied until I saw the advertisements for a new program, Appleworks V1.2. I threw the dinosaur in the closet and acquired the newest in word processing, data base, and spreadsheet all in one. I thought, 'word processing could not get any better!'. Again I was satisfied. In a constant practice to stay with the state of the art, I purchased the new mouse for the Apple //c and soon also, I acquired the Unidisk 3.5 inch drive which I used more and more frequently. Things darkened, I was no longer satisfied. Appleworks could not be used with the convenient mouse and only files could be saved on the Unidisk...not my pre-version 1.3 Appleworks program itself. I hated having to change disks and the Unidisk would have made my job easier. I became depressed.

Finally one day, I was reading my usual stockpile of magazines and noticed an ad for Mouseword, a word processor designed to be used with the mouse. I threw...well...I placed Appleworks in my top ten usefulness rack because it was still powerful, but my word processing needs turned to Mouseword for the most part. Again, I became satisfied. I thought my search was finally over for the perfect word processor.

However, every silver lining is surrounded by a black cloud. All of my reports lacked something. They looked...bland. And having friends who constantly plagued me with 'Macintosh-ized' reports didn't do much for my morale either. My life was over. I had spent hundreds of dollars trying to get top of the line equipment, and for what? One character set, and bold and underlining if I'm lucky. Again I left my computer and read to forget. Then I looked at an ad for Multiscribe. I rubbed my eyes. 'Could it be true?' I thought? I had to see. I ordered away for it and received the package promptly. I had hoped my life would once again have meaning. Little did I know the treat that was to come.

## The Treat That Came

Multiscribe is a new word processing marvel designed for the Apple //c or the enhanced //e. It is a product by Styleware, Inc. (a new company, to the best of my knowledge). And as for their first product, I feel they did remarkably well. Multiscribe is a double hi-res, multi-font word processing system. When I first received the disk, it was copy-protected and contained several serious bugs. I soon lost interest in something that gave me so much problem.

But, Styleware won a place in my heart again...they promptly sent a replacement disk, free of charge, to my door. This replacement not only had the bugs taken out of it, but they also had removed the copy-protection so a person could use it from a hard disk or the Unidisk 3.5. Multiscribe is designed to live with the mouse in perfect harmony, but it is not required. Absolutely all the commands can be done directly from the keyboard. The layout is the classic mouse-style. It uses the window with the elevators. The pull down menus are expertly done and show the keyboard representation right on the menu so you don't have to look every command up in a book. Multiscribe also uses the classic cut and paste techniques. It has mouse-driven find, and find-and-replace utilities and you can easily 'click' your decision for partial word or Whole, Case or no-case.

One of its really interesting features is its graphic representations of 'rulers' that you can insert and delete anywhere in your text. These rulers define indentation, justification, line spacing, left and right margins, and all in a classic typewriter style. And with a simple command, you can show or hide the rulers at any time.

The whole ball of wax is rolled up in the fonts. There are ten fonts including the normal everyday (bland) standard you're probably used to. These fonts are given the names of the people whose styles they are to represent such as Hemingway, Chaucer, Shakespeare, and Asimov. There is even a graphic icon font under the name Michelangelo for those who want their reports spiced up with icons. And all of these fonts are interchangeable in your text with no limit as to the complexity. You could even have separate letters, each in a different font. And all of these fonts can also be changed in size in different arrangements. The sizes range from understated to unbelievable in proportion.

Another feature is the style in which these fonts can be typed. Among these are bold, underline, italic, shadow, outline, subscript, and superscript. And amazingly, you can mix and match these features. This could result in a bold word, underlined and shadowed, superscripted above the rest of the text. The menu for this being the most eye pleasing, because it gives an actual sample for each feature right on the menu. (just try to get that on Mouseword!)

Multiscribe's filing features are also above average. Retrieving files is simple and can all be done with the mouse accepting pathnames and drive numbers. Saving is efficient. You can save a file as it is under a name, or save it under different names at the same time, live. Or you can even save as simple text that can be checked by a spelling checker or whatever.

Multiscribe supports several different printers and printer interfaces, as long as they support graphics capability. A nice touch is when you print out your FIRST document...the printer setup menu comes up automatically before it prints and saves your printer from a garbage attack. Once it is initialized for your printer, or if you already had initialized it for your printer, you will never see the menu again unless you want to.

The printer options menu soon follows and Multiscribe packs a lot of information into one

# a review

window. The options include whether or not you want page numbers and where you want them, how many copies, page range to print, page number to start at, and the quality you want it printed at. This last feature allows you to print your document in ordinary, bland text form, or a preliminary draft form for seeing how it looks. From there you can go to standard quality which is really the double-density option on your printer. And if you choose high quality, Multiscribe prints your file in the little used, mystical, unknown, quad-density option of your printer. This option is obviously used to astound your friends, confuse those who want letter quality, or just to wear out your ribbon. The higher quality you go, the greater time it takes to print your document.

Multiscribe has a memory restriction of 29K or about 9-12 pages depending on what font you use, etc. This is no problem because Multiscribe gives a 'print-merge' command which allows you to print multiple documents with a keypress.

Word processing itself becomes fun. Multiscribe has many of the features more complex processors contain and of course, all of the basic everyday features such as word-wraparound. Multiscribe furthers the quality by providing proportional spaced printing on all your documents. No calculating of page breaks either, you see them pass as you type. The elevator on the side of the typing window shows the current page number. And if you double or triple space a document, you see it happen live! You can even type in double space or triple, you don't simply have to take the computer's word for it. When selecting text with the mouse, another feature shows up. Just dragging the mouse off the window drags the entire document within reach by scrolling, allowing you to do global changes. You could even change a 12 page document from one font to another with a couple of clicks on a mouse. You can scroll anywhere with the mouse using the elevator or the powerful 'goto-page' feature that doesn't leave you guessing like the elevator tends to. All in all, the processor in its simplest form is one of the easiest to use that I have seen so far.

And the features of Multiscribe do not end right there. Amazingly, Multiscribe also includes a powerful font editor in its package. The editor is separate but easily accessible from Multiscribe and vice versa. The editor is also mouse oriented and uses windows and pull down menus exactly like Multiscribe does. One can edit existing fonts or create their own, easily. It's a treat to use the capabilities of double hi-res. All one has to do is choose a letter to edit. You can select it by pressing it, or by using the mouse and choosing the letter off the superimposed keyboard on the screen. The character is then displayed in fat bits (giant size) on the screen.

Here it can be edited with the mouse or keyboard. You can define the size of the character by adding rows or columns of extra space. And if you decide your creation would look better upside-down, you can flip your product vertically or horizontally.

While you are doing this, six little windows on the screen show live the character you deal with in each of the six styles- plain, bold, underlined, italic, shadowed, and outline. In yet another window, the entire character set is displayed and constantly updated according to what you change. It is as easy to use as Multiscribe itself and could have easily been sold as a separate product because of its professional quality. And when you are done, you can go back to Multiscribe directly from the font editor.

Sure, there are other little features such as inverting the screen for those who like white text on a black screen, but I won't dwell too heavily on them.

As all good programs usually do, Multiscribe contains quirks. Its operating bugs are taken out, but little annoyances are there to stay. They are nothing earth shattering, but are worth mentioning. First of all, give second thought before you get Multiscribe if you easily type in excess of 85 words per minute. The program is slow in wraparound depending on the font and the size. Also, Multiscribe must analyze the data when retrieving it from disk and thus, it takes a little time between when the disk drive stops and when your text appears on the screen. With a little patience, this problem is quickly obliterated.

There are many other functions such as a clock, puzzle, calendar, calculator, and a mysterious control panel which does not work when you purchase the program. It has been advertised that these things are an add-on, but my clock card does not activate the clock, and Styleware gives us no hint as to what to do to get this upgraded. Multiscribe does not let you change top or bottom margins, either. Although it is powerful, it is simply not a full featured word processor. Your best bet in this case would be to make your text with Appleworks or some other advanced word processor and then load the file into Multiscribe to spice it up. Also, some commands require data off the disk and this can slow things down in the running of the program.

One of its biggest drawbacks is the documentation. It's a shame that the documentation could not be as of high quality as the program. The documentation is sketchy and sometimes crude. The information you are looking for may be there, but it's probably spread around so much between other material that you could get frustrated. Also, I noticed no mention of the clock, puzzle, calculator, calendar, or control panel in the manual. It never even mentioned where you could get an update for these things. But really, the manual was the only thing of the entire package that left me wanting more.

In conclusion, I don't want to leave you with negative feelings about the program. It is one of the easiest word processors to use in spite of its abilities. Multiscribe certainly isn't a program you can simply boot up and create with from day one, but I promise that a few hours with the program and the documentation and you will be on solo missions in no time. I highly recommend Multiscribe to those who want to get more out of their Apple than a mere typewriter could provide.

And the best point of all...the price. List price is $59.95. In a world where you could pay hundreds of dollars for a good word processor, it's nice to know you don't have to!

# Shape Magic

by Burton Lo

**Requirements:**
Apple ][
BASIC programming experience

Are you an Applesoft programmer whose exhausted of coding shape tables by hand? Well, if you are, this program is the solution to your problem. Shape Magic allows you to draw your shape on the screen so you could see what your shape will look like in XDRAW form. Shape Magic will code your shape after you finish drawing it and allow you to test it in XDRAW form. Shape Magic also allows you to save individual shapes on disk for later use. You can even combine individually saved shapes into a shape table.

Shape Magic doesn't require you to buy any special hardware or software. All you need for using Shape Magic is an Apple monitor or T.V. set for display, 1 disk drive for storage, and the program "Shape Magic." A printer is useful (for printing the codes) but not necessary.

## Using Shape Magic

To use Shape Magic, simply load the program and select an option from the menu by pressing the letter inside the parentheses. This program responds to only uppercase letters, so make sure that the caps Lock (if you have a //e) is in its lower position.

### Menu

If you press **C** on the menu, your computer will clear Hi-Res screen page 1, and draws a line around the screen. This line doesn't mean that you can't go past it, it is just there to let you know where the edge of the screen is. Then you have an option of drawing your shape in regular size or enlarged size. Press "E" for enlarged size, or "R" for regular size. After that, your computer will display a cursor in the middle of the screen in the size you have selected. You are allowed to draw a shape with up to 500 moves. The controls for this section are listed below:

| Key | Purpose |
|---|---|
| E | move up without plot |
| F | move right without plot |
| D | move down without plot |
| S | move left without plot |
| I | move up with plot |
| L | move right with plot |
| K | move down with plot |
| J | move left with plot |
| RETURN | for end of shape, press this key when you have finished your shape |
| <- | go back one move, used for corrections |
| ESC | exit to menu. Note: shape you were working on will be erased! |
| Space | toggle 4 lines of text & graphics |

After you finished drawing your shape, press Return and you will be transferred to another section for printing the codes. This is the same part of the program that gets called when you press **P** from the menu. At this point, your computer will ask if you want printer output. What it really is asking is if you want the shape codes to be printed on your printer. Press "Y" for yes, or "N" for no. If you pressed "Y", your computer will ask you for the name of the shape and slot with your printer interface. Simply type in the answer and press Return. Be sure your printer is on. The codes will then be listed on the screen and, if you selected the printer option, the printer as well. While the codes are being listed, they are also being POKEd into memory.

If you press **D** on the menu, you will be transferred to the DOS commands menu. We will discuss more about this menu later on in this manuscript.

If you press **T** on the menu, your computer will clear Hi-Res screen page 1, and draw a line around the edge of the screen. It you have more than one shape in memory, your computer will ask for the number of the shape that is in memory you wanted to test. Notice that your computer will not except illegal quantity shape numbers. Next, your computer will XDRAW the shape on the center of the screen, and wait for your command. The command keys are listed below:

| Key | Purpose |
|---|---|
| I | move shape up |
| J | move shape left |
| K | move shape down |
| L | move shape right |
| S | increase SCALE (size) |
| R | increase ROT (rotation) |
| RETURN | return to default ROT and SCALE |
| ESC | return to menu |

If you press the **ESC** key on the menu, the program will end and the present shape(s) in memory will be cleared.

## DOS Commands

We will now discuss the DOS commands menu. It is actually a selection of commands which enables you to access your disk drive. You can make a selection simply by pressing a letter inside the parentheses.

If you press **C** in the DOS commands menu, your disk CATALOG will be displayed.

If you press **L** in the DOS commands menu, your computer will ask you for the name of the shape you wanted to load. Type the name of the shape you wanted to load and press Return. If you change your mind about loading a shape, just press Return with no filename.

***Note:*** *When you CATALOG your disk, you will see your shape files preceded by "SH." and your shape tables preceded by "TB.". When you are entering shape names or shape tables, you don't need to enter "SH." or "TB." because this program will do that for you.*

After the shape has been loaded, your computer will transfer you to the Print Codes section, and return you to the main Menu after it has finished listing the codes.

If you press **M** in the DOS commands menu, your computer will ask you to enter a name that is on disk for each shape number starting with 1. Enter a shape name and press Return. The program will ask you for up to 255 shapes. If at any time you want to stop, just type Return with no filename. Shape Magic will then list the shape names of the table and wait for you to make changes.

If you want to replace a shape, simply type the number of the old shape name, and then type in the new name. If you want to remove a certain shape, enter the number of the shape and a blank for the new shape.

After you have finished making changes, enter "0" for no change and press Return. Then you will be asked for the name of the shape table. If you type Return with no filename, you will be allowed to make more changes. This program will not except existing shape table names so you won't lose an old shape table.

Next, your computer will start loading shapes and put them in memory. After that, it will then save the shape table to your disk. You can also test the shapes in the shape table after it have been saved. This can be done only if you haven't put a new shape in memory yet, and you must select Test Shape from menu.

If you press **S** in the DOS commands menu, your computer will ask you for the name of the shape. Enter the name of the shape and press Return. If you change your mind about saving a shape, enter a blank for shape name and your computer will return you to the DOS commands menu. Be sure to enter a name that is not on disk because this section of the program will not check for existing files. If you enter a name that is already on disk, the old shape will be replaced by the new.

If you press ESC in the DOS commands menu, you will be returned back to the main menu. Be sure to check to see which menu you are in before you make a selection because both menu looks very similar to each other, and some selections use the same letter in different a menu.

### How To Use Shape Table

Shape tables are a special way to draw Hi-Res pictures on your Apple ][ computer. It is better than HPLOT because it has special functions such as increasing the size of your picture, rotate your picture, and it enables you to remove your picture from the screen without affecting the background.

After your shape table has been saved, you can retrieve and use the shapes in the shape table and draw them in your Hi-Res graphics screen with the following direct commands:

```
BLOAD TB.EXAMPLE
POKE 233,64 : POKE 232,0
SCALE = 1
ROT = 0
HGR
XDRAW # AT X,Y
```

The 2 POKE commands tell your computer where the shape table starts. Memory location 232 contains the last 2 hexadecimal digits, and memory location 233 contains the first 2 hexadecimal digits. Refer to your reference manual for help with hexadecimal.

The "#" in the above example must be replaced with the number of the shape to draw. It can be any legal Applesoft expression such as a variable.

All of the above commands can also be used in your own programs. Be sure that the first 4 commands are executed before you start drawing any shapes. If you are going to use the shape table in your own program, the first command "BLOAD TB.EXAMPLE" should be changed to the follow:

```
10 PRINT CHR$( 4); "BLOAD
   TB.EXAMPLE"
```

*Note: The above line does not have to be the line 10 in your program, but it should be somewhere in the beginning of your program.*

If you use the above command, do not use HGR2 because it loads the shape table into memory locations where Hi-Res page 2 is located. If you want to use the above line and HGR2 in your program, you should relocate the shape table when you load it. For example, if you wish to put the shape table in memory locations starting with 24576, use the following line instead of the above line:

```
10 PRINT CHR$( 4); "BLOAD
TB.EXAMPLE,A24576"
```

And use the following POKE values in place of the earlier POKE values:

```
20 POKE 232,0: POKE 233,96
```

### Typing In the Program

There is only one listing for this program, so simply type in the listing line by line and save it under any name you wish to call it.

### How Shape Magic Works

| line #: | Purpose: |
|---|---|
| 10 - 50 | Identifies this program |
| 60 - 70 | Sets aside memory for program and dimensions some major variables |
| 80 - 110 | Puts the title of the program at the top of the screen |
| 110 - 120 | Tells the user that we are storing data and sets defaults |
| 130 | Stores a shape table at $300 with 2 shapes in it |
| 140 - 250 | Main menu. redirects program depending upon key pressed |
| 260 - 270 | Subroutine to center A$ on the current line |
| 280 - 360 | Used while drawing a shape to go back one move |
| 370 - 490 | Subroutine for determining position of cursor |
| 500 - 650 | Error handling routines |
| 660 - 670 | Toggles graphic and text page |
| 680 - 930 | Draws shape |
| 940 - 990 | Codes shape codes in decimal form |
| 1000 - 1170 | Lists shape codes |
| 1100 - 1140 | Converts decimal to hexadecimal and pokes value into shape table |
| 1180 - 1320 | The test shape code |
| 1330 - 1430 | DOS commands menu. Program is redirected depending upon keypress |
| 1440 - 1460 | Displays CATALOG of disk currently in drive |
| 1470 - 1560 | Loads shape codes menu selection |
| 1570 - 2020 | Make shape tables menu selection |
| 1590 - 1610 | Gets shape name from user |
| 1620 - 1650 | Checks for shape existence on disk and that it is a text file |
| 1660 - 1680 | List shape names for shape table to be made |
| 1690 - 1720 | Allows the user to make changes to the shape list |
| 1730 - 1750 | Verifies existance of the new shape to be substituted |
| 1760 - 1790 | Removes the shape specified from the list of shapes |
| 1800 - 1810 | Gets the name of the shape table to be made from the user |
| 1820 - 1850 | Makes sure that the filname provided doesn't already exist |
| 1860 - 1890 | Prepares memory for shape table and tells user to wait |
| 1900 - 1980 | Loads a shape and POKEs its data into memory |
| 1990 | BSAVEs the newly created shape table |
| 2000 - 2020 | Displays shape table status and waits for a keypress |
| 2030 - 2110 | Saves shape codes of a particular shape |
| 2120 - 2130 | DATA for a shape table of the two cursors used in shape making |

### Suggestions

This program is designed in a way so that different computers having their printer interface in various slots can be used. You can save the trouble of entering the slot number each time you wish to have a hard copy of the codes by changing line 1040 to something like:

```
990 S = 1
```

Where the value on the right side of the equals sign is the slot number of your printer interface card.

Besides making the above modification, you can also add more options to the menu. And I am sure you can think of a lot of them. I hope you have as much fun with this program as I had creating it.

### Shape Magic Program

```
10 REM /\/\/\/\/\/\/\/\/\/\/\
20 REM <  SHAPE MAGIC          >
30 REM <         BY BURTON LO  >
40 REM \/\/\/\/\/\/\/\/\/\/\/
50 REM
60 LOMEM: 24576
70 DIM C(500 ),G(400 ),T$(255 )
80 TEXT : HOME : INVERSE : VTAB 1 : PRINT SPC(
    40 ) : VTAB 7 : PRINT SPC( 40 )
90 VTAB 2 : FOR I = 1 TO 5 : HTAB 1 : PRINT "^ "
    ; : HTAB 40 : PRINT "^ " ; : NEXT
100 FLASH :A$ = "SHAPE^ MAGIC" : VTAB 3 :
    GOSUB 270
110 NORMAL :A$ = "(C)1986" : VTAB 5 : GOSUB
    270 :A$ = "STORING^ DATA"
120 VTAB 16 : GOSUB 270 : POKE 34 ,7 : POKE 232
    ,0 : POKE 233 ,3 : SCALE= 1
130 FOR ML = 768 TO 789 : READ MN : POKE ML ,MN
    : NEXT ML : ROT= 0
140 HOME : PRINT :A$ = "MENU" : INVERSE :
    GOSUB 270 : NORMAL
150 PRINT : PRINT "(C)REATE^ SHAPE"
160 PRINT : PRINT "(D)OS^ COMMANDS"
170 PRINT : PRINT "(P)RINT^ CODES"
180 PRINT : PRINT "(T)EST^ SHAPE"
190 PRINT : PRINT "<ESC>^ FOR^ EXIT"
200 VTAB 20 : WAIT - 16384 ,128 : GET B$ : IF
    B$ = CHR$ (27 ) THEN HOME : POKE 34 ,0 :
    END
210 IF B$ = "C" THEN 680
220 IF B$ = "D" THEN 1330
230 IF B$ = "P" THEN 1000
240 IF B$ = "T" THEN 1180
250 GOTO 200
260 REM CENTER A$ ON LINE
270 HTAB ((40 - LEN (A$ ) ) / 2 ) : PRINT A$ :
    RETURN
280 IF B = 1 THEN POP : GOTO 680
290 HGR : HCOLOR= 3 : HPLOT 0 ,0 TO 279 ,0 TO
    279 ,159 TO 0 ,159 TO 0 ,0
300 D = B - 2 :X = 140 :Y = 80 : XDRAW A AT X ,Y
310 FOR B = 1 TO D : IF C(B ) < 4 THEN XDRAW A AT
    X ,Y
```

## Selected Variables List

| Variable: | Purpose: |
|---|---|
| A$ | string for printing in middle of line |
| B | number of shape moves |
| C() | direction of movement & plot/ no plot |
| ER | error number |
| G() | shape codes |
| T$() | names of shapes in shape table |
| X | horizontal location of shape for drawing |
| Y | vertical location of shape for drawing |

```
320 PRINT TAB( 1 ); "#" ;B; TAB( 20 );C(B ) :
    GOSUB 370
330 XDRAW A AT X ,Y : NEXT B
340 B = D + 1 : XDRAW A AT X ,Y
350 IF B = 1 THEN POP : GOTO 680
360 RETURN
370 IF C(B ) = 0 THEN Y = Y - 1 - 4 * (A > 1 )
380 IF C(B ) = 1 THEN X = X + 1 + 4 * (A > 1 )
390 IF C(B ) = 2 THEN Y = Y + 1 + 4 * (A > 1 )
400 IF C(B ) = 3 THEN X = X - 1 - 4 * (A > 1 )
410 IF C(B ) = 4 THEN Y = Y - 1 - 4 * (A > 1 )
420 IF C(B ) = 5 THEN X = X + 1 + 4 * (A > 1 )
430 IF C(B ) = 6 THEN Y = Y + 1 + 4 * (A > 1 )
440 IF C(B ) = 7 THEN X = X - 1 - 4 * (A > 1 )
450 IF X < 0 THEN X = 280 - 1 - 4 * (A > 1 )
460 IF X > 279 THEN X = 0 + 4 * (A > 1 )
470 IF Y < 0 THEN Y = 160 - 1 - 4 * (A > 1 )
480 IF Y > 159 THEN Y = 0 + 4 * (A > 1 )
490 RETURN
500 REM ERROR TRAPPER
510 ER = PEEK (222 ) :G = 1 : GOSUB 660 : PRINT
    : PRINT "ERROR^ #" ;ER; CHR$ (7 )
520 IF ER = 4 THEN PRINT "DISK^
    WRITE-PROTECTED."
530 IF ER = 8 THEN PRINT "I/O^ ERROR."
540 IF ER = 9 THEN PRINT "DISK^ FULL."
550 PRINT : PRINT "PLEASE^ FIX^ THE^
    PROBLEM^ AND^ PRESS^ ANY^ KEY.^ " ;
560 GET C$ : RESUME
570 REM ALTERNATE ERROR TRAPPER
580 ER = PEEK (222 ) :G = 1 : GOSUB 660 : PRINT
    "ERROR^ #" ;ER; CHR$ (7 )
590 IF ER = 6 THEN PRINT "NO^ SUCH^ SHAPE^
    ON^ DISK." : GOTO 1590
600 IF ER = 8 THEN PRINT "I/O^ ERROR." : GOTO
    620
610 PRINT "UNEXPECTED^ ERROR."
620 PRINT "PLEASE^ FIX^ THE^ PROBLEM^ AND^
    PRESS^ ANY^ KEY.^ " ; : GET C$ : RESUME
630 REM OTHER ALT ERROR TRAPPER
640 ER = PEEK (222 ) :G = 1 : GOSUB 660 : IF ER
    < > 6 THEN 580
650 PRINT "NO^ SUCH^ SHAPE^ ON^ DISK." :
    POKE 216 ,0 : GOTO 1710
660 IF G = 1 THEN G = 0 : POKE - 16303 ,0 :
    RETURN
670 G = 1 : POKE - 16304 ,0 : POKE - 16297 ,0 :
    RETURN
680 HOME :G = 1 : VTAB 24 : POKE 232 ,0 : POKE
    233 ,3
690 HGR : HCOLOR= 3 : HPLOT 0 ,0 TO 279 ,0 TO
    279 ,159 TO 0 ,159 TO 0 ,0
700 PRINT : PRINT "DO^ YOU^ WANT^
    ENLARGEMENT^ OR^ REGULAR^ SIZE?^
    (E/R)" ; : GET B$
710 IF B$ = "E" THEN A = 2 : GOTO 740
720 IF B$ = "R" THEN A = 1 : GOTO 740
730 GOTO 700
740 POKE 232 ,0 : POKE 233 ,3 : PRINT B$ :X =
    140 :Y = 80 :B = 1 : SCALE= 1 : ROT= 0
750 XDRAW A AT X ,Y : PRINT "MOVE^ #" ;B; ":^
    " ;
760 XDRAW A AT X ,Y : FOR I = 1 TO 100 : NEXT I :
    XDRAW A AT X ,Y
770 FOR I = 1 TO 100 : NEXT I : IF PEEK ( -
    16384 ) < 128 THEN 760
780 GET C$ : IF C$ = "E" THEN XDRAW A AT X ,Y
    :C(B ) = 0 : GOSUB 370 : GOTO 910
790 IF C$ = "F" THEN XDRAW A AT X ,Y :C(B ) = 1
    : GOSUB 370 : GOTO 910
800 IF C$ = "D" THEN XDRAW A AT X ,Y :C(B ) = 2
    : GOSUB 370 : GOTO 910
810 IF C$ = "S" THEN XDRAW A AT X ,Y :C(B ) = 3
    : GOSUB 370 : GOTO 910
820 IF C$ = "I" THEN C(B ) = 4 : GOSUB 370 :
    GOTO 910
830 IF C$ = "L" THEN C(B ) = 5 : GOSUB 370 :
    GOTO 910
840 IF C$ = "K" THEN C(B ) = 6 : GOSUB 370 :
    GOTO 910
850 IF C$ = "J" THEN C(B ) = 7 : GOSUB 370 :
    GOTO 910
860 IF C$ = CHR$ (13 ) THEN 930
870 IF C$ = CHR$ (8 ) THEN GOSUB 280 : GOTO 750
880 IF C$ = CHR$ (27 ) THEN POKE - 16303 ,0 :
    GOTO 140
890 IF C$ = CHR$ (32 ) THEN GOSUB 660
900 GOTO 760
910 PRINT C$; TAB( 20 );C(B ) :B = B + 1 : IF B
    > 500 THEN 930
920 GOTO 750
930 XDRAW A AT X ,Y :G = 1 : GOSUB 660 :B = B - 1
    :E = 0 :F = 1 :G(1 ) = 0
940 A$ = "CODING" : INVERSE : GOSUB 270 :
    NORMAL : FOR I = 1 TO B
950 IF E = 2 AND C(I ) > 0 AND C(I ) < 4 THEN 980
960 IF E < 2 AND (C(I ) > 0 OR C(I ) > 4 ) THEN
    980
970 E = 0 :F = F + 1 :G(F ) = 0
980 G(F ) = G(F ) + C(I ) * (8 ^ E ) :E = E + 1
    : IF E > 2 THEN E = 0 :F = F + 1 :G(F ) = 0
990 NEXT I :F = F + 1 :G(F ) = 0
1000 HOME :S = 0 : PRINT "DO^ YOU^ WANT^
    PRINTER^ OUTPUT?^ (Y/N)" ;
1010 GET C$ : IF C$ = "N" THEN PRINT C$ : GOTO
    1070
1020 IF C$ < > "Y" THEN 1010
1030 PRINT : INPUT "ENTER^ NAME^ OF^ SHAPE:^
    " ;N$
1040 INPUT "ENTER^ #^ OF^ SLOT^ WITH^
    PRINTER^ INTERFACE:^ " ;S : IF S < 1 OR
    S > 7 THEN 1040
1050 PR# S
1060 PRINT "SHAPE^ NAME:^ " ;N$
1070 PRINT "CODE^ #" ; TAB( 10 ); "DECIMAL"
    ; TAB( 20 ); "HEX"
1080 PRINT "------" ; TAB( 10 ); "-------" ;
    TAB( 20 ); "---"
1090 ML = 16384 : POKE ML ,1 : POKE ML + 1 ,0 :
    POKE ML + 2 ,4 : POKE ML + 3 ,0 :ML = ML + 4
1100 FOR I = 1 TO F :Ht = G(I ) / 16 :Lt = G(I ) -
    Ht * 16
1110 Ht = Ht + 176 :Lt = Lt + 176
1120 IF Ht > 185 THEN Ht = Ht + 7
1130 IF Lt > 185 THEN Lt = Lt + 7
1140 M$ = CHR$ (Ht ) + CHR$ (Lt ) : POKE ML ,G(I
    ) :ML = ML + 1
1150 PRINT I; TAB( 10 );G(I ); TAB( 20 );M$
1160 NEXT I : IF S < > 0 THEN PR# 0
1170 PRINT : PRINT "PRESS^ ANY^ KEY.^ " ; :
    GET C$ : GOTO 140
1180 HOME : HGR : HCOLOR= 3 : HPLOT 0 ,0 TO 279
    ,0 TO 279 ,159 TO 0 ,159 TO 0 ,0 :A = 1
1190 VTAB 22 :X = 140 :Y = 80 : IF PEEK (16384
    ) > 1 THEN INPUT "ENTER^ SHAPE^ #:^ "
    ;A$ :A = VAL (A$ ) : IF A < 1 OR A > PEEK
    (16384 ) THEN 1190
1200 POKE 232 ,0 : POKE 233 ,64 : VTAB 24 :
    INVERSE : PRINT TAB( 15 ); "X" ; TAB( 20
    ); "Y" ; TAB( 25 ); "ROT" ; TAB( 30 );
    "SCALE" ; TAB( 39 );
1210 S = 1 :R = 0 : SCALE= S : ROT= R : NORMAL :
    POKE 35 ,23 : XDRAW A AT X ,Y : VTAB 22 :
    PRINT
1220 PRINT "COMMAND?^ " ; : GET C$ : SCALE=
    S : ROT= R : XDRAW A AT X ,Y
1230 IF C$ = "I" THEN Y = Y - 5 : IF Y < 0 THEN Y
    = 155
1240 IF C$ = "J" THEN X = X - 5 : IF X < 0 THEN X
    = 275
1250 IF C$ = "K" THEN Y = Y + 5 : IF Y > 155 THEN
    Y = 0
1260 IF C$ = "L" THEN X = X + 5 : IF X > 275 THEN
    X = 0
1270 IF C$ = "S" THEN S = S + 1 : IF S > 255 THEN
    S = 1
1280 IF C$ = "R" THEN R = R + 1 : IF R > 63 THEN
    R = 0
1290 IF C$ = CHR$ (13 ) THEN 1180
1300 IF C$ = CHR$ (27 ) THEN POKE - 16303 ,0 :
    POKE 35 ,24 : GOTO 140
1310 PRINT C$; TAB( 15 );X; TAB( 20 );Y; TAB(
    25 );R; TAB( 30 );S
1320 SCALE= S : ROT= R : XDRAW A AT X ,Y : GOTO
    1220
1330 HOME :A$ = "DOS^ COMMANDS" : INVERSE :
    GOSUB 270 : NORMAL
1340 D$ = CHR$ (13 ) + CHR$ (4 ) : ONERR GOTO
    510
1350 PRINT : PRINT "(C)ATALOG" : PRINT :
    PRINT "(L)OAD^ SHAPE"
1360 PRINT : PRINT "(M)AKE^ SHAPE^ TABLE" :
    PRINT : PRINT "(S)AVE^ SHAPE"
1370 PRINT : PRINT "<ESC>^ FOR^ EXIT"
1380 VTAB 20 : GET C$ : IF C$ = CHR$ (27 ) THEN
    140
1390 IF C$ = "C" THEN 1440
1400 IF C$ = "L" THEN 1470
1410 IF C$ = "M" THEN 1570
1420 IF C$ = "S" THEN 2030
1430 GOTO 1380
```

```
1440 HOME :A$ = "CATALOG" : INVERSE : GOSUB
     270 : NORMAL
1450 PRINT D$; "CATALOG"
1460 PRINT : PRINT "PRESS^ ANY^ KEY.^ " ; :
     GET C$ : GOTO 1330
1470 HOME :A$ = "LOAD^ SHAPE^ CODES" : GOSUB
     270
1480 PRINT : INPUT "ENTER^ SHAPE^ NAME:^ "
     ;F$
1490 IF F$ = "" THEN 1330
1500 PRINT D$; "VERIFY^ SH." ;F$ : PRINT D$;
     "OPEN^ SH." ;F$
1510 PRINT D$; "READ^ SH." ;F$ :F = 0
1520 F = F + 1 :G(F ) = 0 : INPUT G(F )
1530 IF G(F ) = 0 AND F > 1 THEN 1550
1540 GOTO 1520
1550 PRINT D$; "CLOSE^ SH." ;F$
1560 GOTO 1000
1570 HOME :A$ = "MAKE^ SHAPE^ TABLE" : GOSUB
     270 : I = 0
1580 I = I + 1 :T$(I ) = "" : IF I > 255 THEN I =
     I - 1 : GOTO 1660
1590 PRINT "ENTER^ SHAPE^ #" ;I; ":^ " ; :
     INPUT "" ;T$(I )
1600 IF T$(I ) = "" AND I > 1 THEN I = I - 1 :
     GOTO 1660
1610 IF I = 1 AND T$(I ) = "" THEN 1330
1620 ONERR GOTO 580
1630 PRINT D$; "VERIFY^ SH." ;T$(I ) : PRINT
     D$; "OPEN^ SH." ;T$(I )
1640 PRINT D$; "CLOSE^ SH." ;T$(I )
1650 GOTO 1580
1660 HOME : PRINT TAB( 1 ); "SHAPE^ #" ; TAB(
     10 ); "SHAPE^ NAME"
1670 PRINT TAB( 1 ); "--------" ; TAB( 10 );
     "----------"
1680 FOR J = 1 TO I : PRINT TAB( 1 );J; TAB( 10
     );T$(J ) : NEXT J
1690 PRINT : INPUT "ENTER^ NUMBER^ TO^
     CHANGE^ (0^ =^ NO^ CHANGE):^ " ;K$ :K =
     VAL (K$ )
1700 IF K < 1 OR K > I THEN 1800
1710 INPUT "ENTER^ NAME^ OF^ NEW^ SHAPE:^ "
     ;T$(K )
1720 IF T$(K ) = "" THEN 1760
1730 ONERR GOTO 640
1740 PRINT D$; "VERIFY^ SH." ;T$(K ) : PRINT
     D$; "OPEN^ SH." ;T$(K )
1750 PRINT D$; "CLOSE^ SH." ;T$(K ) : GOTO
     1660
1760 I = I - 1 : FOR J = K TO I
1770 T$(J ) = T$(J + 1 ) : NEXT J
1780 IF I < 1 THEN 1570
1790 GOTO 1660
1800 PRINT : INPUT "ENTER^ SHAPE^ TABLE^
     NAME:^ " ;N$
1810 IF N$ = "" THEN 1660
1820 ONERR GOTO 1850
1830 PRINT D$; "VERIFY^ TB." ;N$
1840 PRINT "EXISTING^ SHAPE^ TABLE.^ ENTER^
     NEW^ NAME." ; GOTO 1690
1850 IF PEEK (222 ) < > 6 THEN 510
1860 POKE 216 ,0 : HOME : INVERSE :A$ =
     "SHAPE^ TABLE:^ " + N$
1870 VTAB 14 : GOSUB 270 :A$ = "PLEASE^
     WAIT!" : VTAB 18 : GOSUB 270
1880 NORMAL :ML = 16384 : POKE ML ,I : POKE ML
     + 1 ,0
1890 MM = ML + 2 * (I + 1 ) :MN = ML + 2
1900 FOR L = 1 TO I : ONERR GOTO 510
1910 Q = MM - ML : IF Q > 255 THEN 1930
1920 POKE MN ,Q : POKE MN + 1 ,0 : GOTO 1940
1930 U = INT (Q / 256 ) :V = Q - U * 256 : POKE
     MN ,V : POKE MN + 1 ,U
1940 MN = MN + 2 : PRINT D$; "OPEN^ SH." ;T$(L
     ) :T = 1
1950 PRINT D$; "READ^ SH." ;T$(L ) : INPUT
     G$ :G = VAL (G$ )
1960 POKE MM ,G :MM = MM + 1 : IF G = 0 AND T > 1
     THEN 1980
1970 T = T + 1 : GOTO 1950
1980 PRINT D$; "CLOSE^ SH." ;T$(L ) : GOSUB
     270 : NEXT L
1990 PRINT D$; "BSAVE^ TB." ;N$; ",A" ;ML;
     ",L" ;MM - ML
2000 PRINT : PRINT "SHAPE^ TABLE:^ " ;N$ :
     PRINT "STARTING^ ADDRESS:^ " ;ML
2010 PRINT "LENGTH:^ " ;MM - ML
2020 PRINT "PRESS^ ANY^ KEY.^ " ; : GET C$ :
     GOTO 1330
2030 HOME :A$ = "SAVE^ SHAPE^ CODES" : GOSUB
     270
2040 PRINT : INPUT "ENTER^ SHAPE^ NAME:^ "
     ;F$
2050 IF F$ = "" THEN 1330
2060 IF G(1 ) = 0 AND G(2 ) = 0 THEN PRINT
     "BAD^ SHAPE" : GOTO 2110
2070 PRINT D$; "OPEN^ SH." ;F$ : PRINT D$;
     "DELETE^ SH." ;F$
2080 PRINT D$; "OPEN^ SH." ;F$ : PRINT D$;
     "WRITE^ SH." ;F$
2090 FOR I = 1 TO F : PRINT G(I ) : NEXT I
2100 PRINT D$; "CLOSE^ SH." ;F$
2110 PRINT "PRESS^ ANY^ KEY.^ " ; : GET C$ :
     GOTO 1330
2120 DATA ^ 2 ,0 ,6 ,0 ,8 ,0 ,4 ,0 ,45 ,45
     ,62 ,63
2130 DATA ^ 55 ,45 ,45 ,62 ,63 ,55 ,45 ,45
     ,4 ,0
```

## checksums

```
10   - $BADD    1080 - $92D4
20   - $9B13    1090 - $7121
30   - $4D3B    1100 - $7B91
40   - $AD92    1110 - $B6BE
50   - $C899    1120 - $BB4B
60   - $12CC    1130 - $76F5
70   - $1F00    1140 - $0651
80   - $0CF2    1150 - $4CA3
90   - $F15F    1160 - $6E15
100  - $0C7B    1170 - $A8CB
110  - $8BEC    1180 - $E069
120  - $1787    1190 - $CDD6
130  - $1028    1200 - $055B
140  - $40EC    1210 - $2D46
150  - $A356    1220 - $DEB1
160  - $C1FA    1230 - $08A8
170  - $F25C    1240 - $20F5
180  - $35D6    1250 - $42EB
190  - $4B22    1260 - $D425
200  - $ABE2    1270 - $3C56
210  - $ED72    1280 - $A653
220  - $357B    1290 - $7C5E
230  - $E8F2    1300 - $4CEA
240  - $3D5B    1310 - $5428
250  - $4FFF    1320 - $B926
260  - $DB5F    1330 - $8DAE
270  - $F428    1340 - $EE91
280  - $9DEE    1350 - $5CDF
290  - $98B7    1360 - $120C
300  - $0B37    1370 - $71B6
310  - $B6ED    1380 - $74C5
320  - $7005    1390 - $4E6B
330  - $0428    1400 - $F589
340  - $EC80    1410 - $A34F
350  - $E379    1420 - $FC0D
360  - $9620    1430 - $E32F
370  - $C898    1440 - $FF2C
380  - $C349    1450 - $A60A
390  - $5374    1460 - $1864
400  - $CD6D    1470 - $109E
410  - $95BF    1480 - $7E27
420  - $D484    1490 - $9636
430  - $8EBF    1500 - $ACDC
440  - $F680    1510 - $D27C
450  - $1262    1520 - $DA5A
460  - $AB8F    1530 - $7F78
470  - $13DA    1540 - $FCBD
480  - $444C    1550 - $4EF1
490  - $4734    1560 - $358A
500  - $BC79    1570 - $B837
510  - $A58F    1580 - $71F1
520  - $C292    1590 - $C7C5
530  - $4848    1600 - $B293
540  - $1FFB    1610 - $ADF1
550  - $769B    1620 - $F35C
560  - $0617    1630 - $EC92
570  - $2482    1640 - $D75D
580  - $104E    1650 - $583E
590  - $BFD1    1660 - $1F82
600  - $70E2    1670 - $297D
610  - $BA45    1680 - $1F9D
620  - $AE69    1690 - $DDA6
630  - $D2C8    1700 - $EA90
640  - $D947    1710 - $5302
650  - $EFAD    1720 - $8846
660  - $ACDC    1730 - $E7D7
670  - $874A    1740 - $74B0
680  - $2D4B    1750 - $F5A3
690  - $6EE6    1760 - $28D4
700  - $DA63    1770 - $07DB
710  - $A95A    1780 - $0631
720  - $7D11    1790 - $6248
730  - $FC80    1800 - $8049
740  - $785D    1810 - $302D
750  - $3377    1820 - $9040
760  - $A40B    1830 - $3681
770  - $C70E    1840 - $6E45
780  - $6B2D    1850 - $C4E0
790  - $AA47    1860 - $5CDA
800  - $7C88    1870 - $20C6
810  - $B55E    1880 - $3EA3
820  - $4831    1890 - $50D1
830  - $EEB4    1900 - $9D16
840  - $7D81    1910 - $5DDE
850  - $A178    1920 - $A346
860  - $43AE    1930 - $B34D
870  - $65E8    1940 - $7F42
880  - $F743    1950 - $6C84
890  - $5BE9    1960 - $DC4F
900  - $CF24    1970 - $C0E3
910  - $17ED    1980 - $8271
920  - $A2B7    1990 - $287C
930  - $33F3    2000 - $C2AB
940  - $F362    2010 - $8B19
950  - $9116    2020 - $EDB6
960  - $C65F    2030 - $0264
970  - $2987    2040 - $84D0
980  - $7BBE    2050 - $8F71
990  - $305F    2060 - $A518
1000 - $7FF9    2070 - $E9EF
1010 - $737B    2080 - $25B3
1020 - $D659    2090 - $BD04
1030 - $8065    2100 - $5257
1040 - $BA56    2110 - $A695
1050 - $B91F    2120 - $F006
1060 - $FAAD    2130 - $6FDF
1070 - $3BA0
```

# Amazing Computer Facts

by Rich Etarip

I have come up with several neat programs, tricks and modifications for the Apple that can be quite interesting and fun. In this article I will show you the following:

1) How unbelievably fast machine language is

2) How to get rid of that annoying grinding your disk drive makes upon encountering an I/O ERROR

3) How to possibly increase the running speed of a video game

4) A way to encode an assembly language JMP

5) How to BSAVE a binary file without typing the A$,L$ parameters.

6) How to encode an assembly language routine so it can't be read.

First lets start with number 1. As everyone knows, machine language is the native tongue of the Apple. Any other language you program in must be translated to something the computer can understand before it can be executed. As an example, imagine you were trying to speak to a Spanish person but you didn't know a word of Spanish. All you had was an English/Spanish conversion dictionary. It would take you almost forever to translate each individual word to spanish as you were speaking than if you knew how to speak Spanish. However, this is BASICally (no pun intended) how Applesoft BASIC works. Each instuction must first be translated to machine language by the BASIC interpreter before it can be executed. That is why BASIC runs so slow. Wouldn't it be a treat if machine language was as easy to program in as BASIC?

Now let's get to my example. First take out your Apple master diskette and run the program called COLOR DEMOSOFT. Next, choose option 3, the kaleidoscope. I really liked the colorful design of this kaleidoscope but it was just too slow so I decided to do something about it. I simply sat down and translated the program, one instruction at a time, to machine language. I could not beleive the difference in speed from BASIC to machine language. It took BASIC about 21 minutes to complete the entire program where it took machine language only 30 seconds. That's about 42 times faster than BASIC. Now key in the following listing at $1000:

```
1000: 4C 20 10 A6 00 A9 00 18    $3535
1008: 65 01 CA D0 FA 60 A2 00    $8F4E
1010: A5 00 38 E5 01 90 06 85    $62FC
1018: 00 E8 4C 10 10 8A 60 00    $645D
1020: 20 58 FC 20 40 FB A9 03    $C26E
1028: 85 02 A9 01 85 03 A9 00    $CA80
1030: 85 04 A5 03 18 65 04 85    $8941
1038: 05 A5 04 85 00 A9 03 85    $FE71
1040: 01 20 03 10 85 06 A5 03    $9C41
1048: 18 69 03 85 07 A5 03 85    $C356

1050: 00 A5 02 85 01 20 03 10    $7DF8
1058: 85 08 85 00 A9 0C 85 01    $7A02
1060: 20 0E 10 85 09 A5 00 85    $DF18
1068: 0B A5 06 85 00 A5 07 85    $1440
1070: 01 20 0E 10 85 0A A5 00    $21E6
1078: 18 65 0B C5 03 90 02 E6    $F45F
1080: 0A A5 0A 18 65 09 20 64    $3EF8
1088: F8 A4 03 A5 05 20 00 F8    $3AB6
1090: A4 05 A5 03 20 00 F8 A9    $2B16
1098: 28 38 E5 03 85 0B A9 28    $A9BD

10A0: 38 E5 05 85 0C A4 0B A5    $7D1D
10A8: 0C 20 00 F8 A4 0C A5 0B    $EA7F
10B0: 20 00 F8 A4 05 A5 0B 20    $861B
10B8: 00 F8 A4 0B A5 05 20 00    $DCC7
10C0: F8 A4 03 A5 0C 20 00 F8    $A49C
10C8: A4 0C A5 03 20 00 F8 E6    $CB5D
10D0: 04 A5 04 C9 14 D0 11 E6    $399F
10D8: 03 A5 03 C9 14 D0 0C E6    $FD42
10E0: 02 A5 02 C9 33 D0 07 60    $A99E
10E8: 4C 32 10 4C 2E 10 4C 2A    $D4B0

10F0: 10                         $085C
```

Once the listing is entered type 1000G to run the program and you will see what I mean. If you don't know how to program in machine language yet, I highly reccommend that you learn it. I've had a lot of fun with it since I learned.

## Sounds from Heaven

How many of you can't stand that obnoxious grinding sound your disk drive makes when it can't read. I would like it if DOS would tell me kindly when it has run into a problem rather than barking at me and letting all the neighbors know that I've run into an I/O ERROR (I have a very loud disk drive). Somebody who doesn't know that it is the disk drive making that noise may think that I own a mad dog.

When DOS encounters a problem, it recalibrates the arm of the disk drive back to track 0 and when it reaches track 0 and tries to go further, the disk drive makes the grinding sound because track 0 is as far as it can go. I looked through DOS and figured out where it was coming from. At $BDD2 begins the routine that causes the disk drive to grind. I made a modification to this to remove the sound and everything seems to work just fine. It seems to me that this sound was put into DOS only as an effect. Don't quote me on it because I'm not positive but that's the way it appears. All you have to do is type:

**BDD4:4C 04 BE**

to bypass the routine and you should be free of that ugly sound. For a test, open your disk drive door and try to CATALOG your disk. If if is working correctly, the only thing you should hear is a -BEEP-. If you wish to make this change to the DOS already on your disk, read Track 0 Sector 7 and at byte $D4 enter 4C 04 BE.

## And faster, and faster...

The third neat thing I'd like to show you is how to increase the running speed of some Apple video games. The monitor delay routine

is located at $FCA8. Quite a few video games are slowed down (if they run too fast) by using this routine. Now if we were to load in a game and search through memory for every JSR to $FCA8 and skip over all of them, the game would run somewhat faster. Of course this all depends upon how much the game was originally slowed down by this delay or if the programmer used his own delay routine. The following program will search memory for calls to $FCA8 and change them all to JSR $FCB3 which is simply the RTS from the routine.

```
1280-   A2 00       LDX   #$00
0282-   BD 00 08    LDA   $0800,X
0285-   C9 A8       CMP   #$A8
0287-   D0 0C       BNE   $0295
0289-   BD 01 08    LDA   $0801,X
028C-   C9 FC       CMP   #$FC
028E-   D0 05       BNE   $0295
0290-   A9 B3       LDA   #$B3
0292-   9D 00 08    STA   $0800,X
0295-   E8          INX
0296-   D0 EA       BNE   $0282
0298-   EE 84 02    INC   $0284
029B-   EE 8B 02    INC   $028B
029E-   EE 94 02    INC   $0294
02A1-   AD 84 02    LDA   $0284
02A4-   C9 96       CMP   #$96
02A6-   D0 DA       BNE   $0282
02A8-   6C 72 AA    JMP   ($AA72)
```

I wrote this program a few years back and tried it on quite a few games. The ones I distinctly remember it having a major effect on were Sneakers and Super Invaders. Super Invaders became virtually unplayable because it ran so fast. Type it in and try it on your games. To save it to disk, type:

**BSAVE DELAY REMOVER,A$280,L$2B**

Here is how to use the program:

1) BLOAD DELAY REMOVER

2) BLOAD (your game)

3) CALL 640 (from BASIC) or 280G (from assembly)

The delay remover program will automatically run the game when it is finished but you MUST load the delay remover BEFORE you load the game or it will not work. Have fun with it!

## Confusion City

There are a number of different ways you can encode a JMP in machine language so a person tracing through your program would have a hard time figuring out where it is going. For instance, if you see a JMP ($0000) where would you think it was jumping? An indirect jump such as this causes a jump to the location stored in $0000 and $0001. For instance, if the values in $00 and $01 are 00 and 40 respectively, that means it will jump to $4000. BUT... What would you think if you saw this JMP ($00FF)? The program will jump to the location stored in $00FF and $0100? WRONG!!

This is a good way to fool somebody because that is what they will think. What will really happen is it will jump to the location stored in $00FF and $0000. This is the infamous JMP bug in the 6502 microprocessor. This bug has been fixed in later releases of the microprocessor (such as the 65C02 and the 65816). Some older programs (like Disk Organizer for example) used this technique as part of their copy protection. They will do a JMP to some ending page address, like $2FF and store the real place they want to jump to at $2FF and $200. Programs that use this will not work on the new //e's and certainly not on the //gs.

## Where Was That File Again?

The next thing I have is a DOS modification. How many times have you BLOADed a file and wanted to save it to another disk so you had to enter the monitor and look up locations $AA72 and $AA73 for the start address and $AA60 and $AA61 for the length? It gets kind annoying to have to do that all the time so I have come up with a short routine you can patch into DOS that will BSAVE your file with the start address and length of the last BLOADed file. If you wish to BSAVE your file with different parameters, simply type them in as you normally would.

Make the following modifications to DOS for this option:

```
A333:20 DF BC
BCDF:A9
BCE0:08 2D 65 AA C9 08 F0 0C
BCE8:AD 60 AA 8D 6C AA AD 61
BCF0:AA 8D 6D AA A9 09 60
```

If you wish to make these changes directly to your disk, edit track 1, sector 2 starting at byte 33 and track 0, sector 6 starting at byte DF.

When a BSAVE command is issued, this routine checks if the A$ and L$ parameters have been typed. The BSAVE command uses the value in location $AA65 to specify what parameters have been typed. Here are the values used:

```
00 - Neither A$ or L$ have been typed
01 - Only A$ has been typed
08 - Only L$ has been typed
09 - Both A$ and L$ have been typed
```

Now, to save a file with the same A and L parameters it was loaded with, you simply type:

**BSAVE filename**

## The Dread Assembly Scramble

The final thing I have is a neat way to scramble assembled machine code so it is not readable, at least not easily readable. All you have to remember is the numbers $07 and $20. Insert these values between each assembly instruction (or every other assembly instruction or however you prefer) and when it is listed, it will come up as junk. Other values besides $07 and $20 will also work, but then again, a lot of them will not. I've seen $04 and $5E used also.

This works best when encoding short routines that you want to hide like nibble count programs because you can't tell what it is by looking. Actually, it doesn't even look like executable machine code. The $07 code is an undefined assembly language instruction and comes up with three question marks when appearing in a listing. Somehow, when machine language encounters this byte, it skips over it and the next byte also. These codes are not used by the 6502 microprocessor and so it just skips over them.

This technique will not work if you try it on an enhanced Apple //e or an Apple //c because the new microprocessor (65C02) has an expanded instruction set. Take a look at the following listing:

```
1000-   20 40 FB    JSR   $FB40
1003-   A9 02       LDA   #$02
1005-   20 64 F8    JSR   $F864
1008-   A0 00       LDY   #$00
100A-   A9 27       LDA   #$27
100C-   85 2C       STA   $2C
100E-   A9 10       LDA   #$10
1010-   20 19 F8    JSR   $F819
1013-   60          RTS
```

This is a simple machine language routine using lo-res commands to draw a horizontal blue line across the screen. I'm sure you wouldn't want bother hiding a program like this but it is just used as an example. Now take a look at this listing:

```
1000-   07          ???
1001-   20 20 40    JSR   $4020
1004-   FB          ???
1005-   07          ???
1006-   20 A9 02    JSR   $02A9
1009-   07          ???
100A-   20 20 64    JSR   $6420
100D-   F8          SED
100E-   A0 00       LDY   #$00
1010-   07          ???
1011-   20 A9 27    JSR   $27A9
1014-   07          ???
1015-   20 85 2C    JSR   $2C85
1018-   07          ???
1019-   20 A9 10    JSR   $10A9
101C-   07          ???
101D-   20 20 19    JSR   $1920
1020-   F8          SED
1021-   07          ???
1022-   20 60 00    JSR   $0060
```

This program functions EXACTLY the same as the previous one. The only difference is that $07 and $20 have been inserted between each instruction. It executes exactly the same except it can't be easily read.

This example uses no internal jumps or branches however. You have to be very careful when using branches within the routine if you are entering it directly into the monitor by hand. Since there is a lot of address changing when you are placing two bytes between each assembly instruction, you have to make sure that the branches and jumps are going to the correct location. However, if you use an assembler, it should be quite simple.

Well, that will about wrap it up. Hope you will have fun with these tricks and modifications and if you come up with any of your own be sure and send them to COMPUTIST.

# Star Blazer

## APT for Lost Tomb

by 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.

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

## Star Blazer

by Rich Etarip

*Broderbund Software*
*17 Paul Drive*
*San Rafael, CA 94903*

**Requirements:**
48K Apple
Initialized disk with no Hello program
Star Blazer diskette

Star Blazer is a fun and quite addictive game from Broderbund Software. Even though it was released quite a few years ago, I still find it fun to play. Being a single load program (that means there is no disk access once the game is in memory) the protection on the disk is quite extensive. The data on the diskette is encoded in 4+4 and every track on the diskette contains different address marks. The boot code on the disk is also quite difficult to trace because it is continuously changing the stack pointer and using the RTS commands for practically all of its jumps. (In case you don't understand what that means, instead of Jumping, the program will push the destination address on the stack and pretend it came from a subroutine. Then an RTS takes the two top values off the stack, adds one, and jumps to that address.) In addition, a theme program is loaded in before the game making things even harder. I even came across a nasty message from the author while tracing through one of the boot stages. It read:

**ARE YOU HAVING FUN? WHY DON'T YOU DO SOMETHING MORE CONSTRUCTIVE, LIKE WRITING A GAME AND MAKE SOME MONEY?**

I always enjoy coming across humorous messages like that while I am trying to deprotect a game. But continuing on...

After countless hours of examining the boot code and disk format I finally managed to deprotect the game via Boot Code Tracing. Here is how it was done:

1) The first thing you have to do is enter the monitor.

```
CALL -151
```

2) Move the boot program from the disk controller card down to RAM so it can be modified.

```
9600<C600.C6FFM
```

3) At $96F8 is a JMP $801 which is the start of the boot stage 1 from track 0 sector 0 of the disk. Change this to jump to $9801 and at $9801 write a routine to turn off the disk drive and jump to the monitor.

```
96FA:98
9801:AD E8 C0 4C 59 FF
```

4) We are ready to execute the boot program at $9600.

```
9600G
```

5) Boot 1 is now loaded in at $800 and we have to move it to $9800 to modify it to read in the next boot stage and return control to us.

```
9800<800.8FFM
```

6) At $9803 is an LDA $0800,X followed by a STA $0200,X. This is relocating boot 1 from $800 to $200. We have to change $9803 to load from $9800 instead because our modified boot 1 is at $9800.

```
9805:98
```

7) If you look at $9837 you will see a JMP $0301. This is the jump to the next boot stage that is loaded in by boot 1. Change this to jump to the monitor instead.

```
9838:59 FF
```

8) Now execute the boot program again.

```
9600G
```

9) The disk drive will still be running so it should be turned off.

```
C0E8
```

10) Next, we have to move boot stage 2 from $300 to $9300 so we can change it.

```
9300<300.3FFM
```

11) Boot 1 is still jumping to $FF59 so change it to jump to the next boot stage at $9301.

```
9838:01 93
```

If you list through this boot stage, you will see readable machine code up until $9326 and the rest appears to be 'garbage' memory... but it really isn't. This boot stage does no disk access at all but rather, decodes the garbage memory into $100 (the stack area) and then exits with an RTS command. By tracing this, I found the RTS 'returning' to $131. You can trace it yourself if you like torture but it would only add extra steps to this already long boot code trace.

12) What we want to do here is write a routine to move page $1 up to page $91 and then jump to the monitor. The place to put it is $9325.

```
9325:BD 00 01 9D 00 91 E8 D0 F7 4C
  59 FF
```

13) Once again, we can execute the boot program.

```
9600G
```

14) Again, turn off the disk drive.

```
C0E8
```

Now let's list through $9100. At $91FB is an LDX $0565 followed by a TSX and a BRK. The first two instructions aren't confusing but that BRK is because it is a forced interrupt. Wait a minute... At $9131 is an LDX #$60 and a STX $01FF. So the BRK at $1FF really gets covered up by a $60 which is an RTS. What the boot stage in page $1 does is load in yet another boot stage from $400 to $7FF and this RTS at $1FF causes a branch to $4A3.

Before writing this article, I examined all of the boot stages hoping to make the boot code trace as short as possible. Because of this, you really won't have to trace through the boot stage at $400. Besides, the text page is a real pain to work with anyway. In this boot stage, pages $5 and $6 are moved down to pages $2 and $3 respectively. Then the theme program from the game is loaded in. The text page boot stage (sounds neat doesn't it) is exited by an indirect jump to $200. Remember that $500 was moved to $200 so what we would need are the values at $500 and $501. I found them to be $00 and $63. First it goes to $6300 and that routine ends in an RTS and where it was going from there didn't matter to me anymore because by snooping through memory I had found the routine that loads the game itself in.

It is loaded into $500 right before the theme program is executed. Well, I thought it was simple. All I had to do was make the boot code jump to the game loading routine instead of the theme program. I tried it and the disk drive just sat there and spun endlessly. With further examination of the boot code, I discovered that the address marks are different on each track and the loading routine looks for them in zero page locations $F0, $F1 and $F2. Now if the proper values are not in those locations, of course the disk drive will not be able to read. There is also another value stored in $F3 that designates the end of a sector. Now the only problem I faced was finding out what these locations should contain. Taking another look at the boot code, I found the values D5 AA D4 B5 being stored in the address mark locations before jumping to the theme program. It seemed obvious that these were the correct values. Now I should have everything I need to know.

15) What we want to do next is write a routine to make two changes to the text page before jumping to it. We could make the changes by hand but that would involve writing a memory move and adding extra steps to the boot code trace. Since all we are changing is two bytes, a simple routine will do the trick easily. The question is where to put the routine. The text page boot is jumped to via an RTS as mentioned earlier. At $91FB is where the stack pointer is changed before the RTS so this is a good place to put a jump to our routine.

```
91FB:4C 00 95
```

16) Location $9500 is a safe place to put this routine. The two bytes we want to change are the $00 at $500 and the $63 at $501 so it doesn't jump to $6300. Instead, let's have it jump to $9510 and where we will write a routine to handle the loading of the game.

```
9500:A9 10 8D 00 05 A9 95 8D 01 05
  AE 65 05 9A 60
```

What this routine does is store $10 and $95 at $500 and $501, then finish what was being done at $91FB.

17) Before we write this routine at $9510 we can't forget about the boot routine at $9300. We never changed it after the last execution and it is still exiting into the monitor. Let's change it to move the modified page $1 (which is at $9100) back down to page $1 and then jump to $131 where it was initially going.

```
9327:91
932A:01
932F:31 01
```

Now we're getting to the meat of the boot code trace. The routine at $9510 must do the following:

A. Store the correct address marks in $F0 through $F3.
B. Patch a 4C 59 FF at the end of the loader routine so it will jump to the monitor when the game is loaded in.
C. Jump to the routine to load in the game at $573.

18) Enter the routine at $9510.

```
9510:A9 D5 85 F0 A9 AA 85 F1
9518:A9 D4 85 F2 A9 B5 85 F3
9520:A9 4C 8D BC 05 A9 59 8D
9528:BD 05 A9 FF 8D BE 05 4C 73 05
```

19) Everything should be set now and we are ready to execute the program at $9600 to load the game in.

```
9600G
```

20) The game should now be loaded in so turn off the disk drive.

```
C0E8
```

As you noticed, the disk drive was still running after the game was loaded in. What this usually means is that there is still more to be loaded in. Since the loader was originally jumping to $600 I took a look at $600 to find that it read in only a few bytes from the disk to make some last minute changes to the zero page. These are vital to the working of the game so I jotted them down for later use. As far as we're concerned, the entire game is loaded in.

21) The game occupies memory from $800 to $A800 with hi-res page 1 ($2000-$3FFF) blank. We want to move page $89 through $A7 down to $2100 to scrunch the file and save page $20 for zero page data and the memory move routine.

```
2100<8900.A7FFM
```

22) What we want to do now is make the changes to the zero page that were mentioned above. We will be moving page $20 down to page $00 during the memory move so the changes should be made to page $20 instead. Before doing that though, move page $00 to page $20 so the zero page pointers are not overwritten during the memory move.

```
2000<00.FFM

2020:D1 17 08 17 1B 17
2028:6C 0D
204E:1C 1D
```

23) Before going any further, move page $8 to $9000 so it will not get overwritten when we reboot DOS.

```
9000<800.8FFM
```

24) Now insert a blank diskette with NO HELLO PROGRAM and reboot DOS.

```
C600G
```

25) Enter the monitor again.

```
CALL -151
```

26) Now move page $8 back where it belongs.

```
800<9000.90FFM
```

27) All that is really left is writing the memory move. Put it at $2080 because we need $2000-$207F for the zero page.

```
2080:20 89 FE 20 93 FE A2 7F
2088:BD 00 20 9D 00 00 CA 10
2090:F7 A2 00 BD 00 21 9D 00
2098:89 E8 D0 F7 EE 95 20 EE
20A0:98 20 AD 95 20 C9 40 D0
20A8:EA A9 99 8D 00 02 A9 19
20B0:8D 01 02 6C 20 00
```

28) Now put a jump to this routine at $7FD right before the start of the program.

```
7FD:4C 80 20
```

29) Finally, patch DOS to save a file longer than $7FFF and save Star Blazer to your disk.

```
A964:FF

BSAVE STAR BLAZER,A$7FD,L$8103
```

You now have Star Blazer deprotected in a BRUNable file. Now dig into the game code and find a way to 'cheat' on it. Have fun!!

## softkey for...

# Science Tool Kit

by Stephen Lau

*Broderbund Software Inc.*
*17 Paul Drive*
*San Rafael, CA 94903*

**Requirements:**
64K Apple II Plus, //e or //c
Super IOB v1.5
a blank disk

Science Tool Kit is a rather new release from Broderbund. Along with the Science Tool Kit master module is some hardware which enables your Apple compatible computer to act as a timer, a chart recorder and more so that you can perform some scientific experiments. The Science Tool Kit enables you to record the results onto a standard disk. Also, optional Science Tool Kit program disks with different kinds of experiments will soon be avaiable, so it seems that this program could be rather valuable to students who take science.

Of course the first thing to do after buying expensive software is to back them up. I was a little bit annoyed when I found that there was no built-in copy program on the disk while the other Broderbund programs (like Fantavision or Print Shop) have, even though they only allow you to make one backup only. So I tried to deprotect it.

Using the CIA, I found that Science Tool Kit occupies the disk from track Ø to track 1F. Track 1 has a very strange format. At first I thought that this track contained no data, but later on I found that I was wrong. The other tracks seem to be encoded in normal DOS 3.3, except that the address trailer has changed from the usual DE AA to DE BB.

This was rather easy to defeat. I immediately wrote a controller to convert them back to normal value. Of course the resulting copy would not boot yet, so I boot code traced the disk.

```
CALL-151
9600<C600.C6F7M
96F8:4C 59 FF
9600G
801L
```

Upon close examination of the Boot1 code of Science Tool Kit, I noticed that the first part of it simply sets up some values in page 3, draws a pattern on the low-res page, displays the low-res page, and then reads three sectors off the disk using the disk controller ROM's subroutine at $C65C, as shown here.

```
83C: LDA #$4C
83E: STA $801
841: LDX $2B    ;load x-register and
843: TXA        ;accumulator
844: LSR        ;with the slot*16 ($60)
845: LSR        ;
846: LSR        ;divide by 2 four times
847: LSR        ;to get a result of 6
848: ORA #$C0   ;OR with C0 to get C6
84A: STA $859   ;store at $859
84D: LDA $85B   ;load data
850: BMI $86B   ;if FF then branch to
852: STA $3D    ;$86B
854: INC $84E   ;point to next sector
857: JMP $Cx5C  ;changed to point to
85A:  00        ; disk II slot ($C65C)
      0D        ;physical sectors to be
      0B        ;loaded
      09
      FF        ;end of table
      ..   ..
      ..   ..
```

This subroutine is of rather great importance in this softkey, you'll see why later. After loading the three sectors into pages $9,$A, and $B, control passes to $86B which sets up some zero page pointers. At $877 and $87C are two JSRs to $900 and $946 respectively. Then it exited through a JMP $1000.

Listing the code at $900 and $946 , I soon found out that $900 is a seek track subroutine, the accumulator holds the value for the track to seek to, the code at $946 checkes for a header of D4 D7 F5, and then using 4+4 encoding, reads a total of six pages into $1000-$15FF. Then control passes to $1000.

So, the folks at Broderbund had made life hard for us, in which the track to be eliminated contains essential code for the rest of the boot process. So we'll have to somehow capture the code from $1000-$15FF, save it to the disk, load the data off the disk at the right moment, and of course we'll have to eliminate the check on track 1.

Now the problem came. Unlike Fantavision, there is no RWTS of any kind loaded into memory at the time the JUMP $1000 is to be taken, so we cannot utilize the Broderbund RWTS to load the code necessary for the rest of the boot. But, remember the code which loads the three sectors off the disk? Upon close examination of track Ø, only (logically numbered) sectors Ø,1,2,3 are used, so we can place the code at track Ø, sectors 4-9, and change the BOOT1 code to load sectors 1-9 into page $9 through page $11. Finally, we'll place a small subroutine at $946 which moves the code from $C00-$11FF to $1000 -$15FF. So let's move.

First the code from $1000-$15FF needs to be captured.

```
CALL-151
9600<C600.C6FFM
96F8:A9 59 8D 80 08 A9 FF 8D 81 08
  4C 01 08
9600G
```

(patch the $1000-$15FF code:)

```
1008:EA EA EA
136C:AA
6(return)P
BSAVE CODE,A$1000,L$600
```

The edits to make on the BOOT1 code are:

```
Add sectors to the loading table:

85E: 7 5 3 1 E C FF

Move the code to its working location:

946: A2 00      LDX #$00
948: BD 00 11   LDA $1100,X
94B: 9D 00 15   STA $1500,X
94E: E8         INX
94F: D0 F7      BNE $948
951: EE 4A 09   DEC $94A
954: EE 4D 09   DEC $94D
957: AD 4A 09   LDA $94A
95A: C9 0B      CMP #$0B
95C: D0 E8      BNE $946
95E: 60         RTS
```

I made these changes to my Super IOBed copy and booted the disk. The disk did boot and I could even get into the menu. But when I tried to do some experiments, I was rewarded with the dreaded message "INSERT YOUR SCIENCE TOOL KIT PROGRAM DISK INTO DRIVE 1". I soon discoverd that during the boot, another RWTS was loaded into memory, which of course checked for the DE BB trailer. I simply normalized it and the program ran.

The disk needs one more sector edit so that it'll have no problems when switching between the normal DOS 3.3 data disk and the program disk (there was a subroutine that changed the trailer back to DE BB after reading a data disk). Then the disk works like a charm.

## About the Controller

The controller is just a modification of the FANTAVISION controller (COMPUTIST No. 20). The data to be written onto track 0 sectors 4-9 are directly loaded into the track buffer ($2B00- $30FF) of Super IOB. After the controller has read track 0, sectors 0-3, into the beginning of the track buffer ($2700-$2AFF), the whole crunch ($2700-$30FF) is written onto the backup. The subroutine at 2000 is just another way of sector editing a disk (apart from that in the IOB), which is more handy than the subroutine already in Super IOB when a series of code has to be patched.

## Step by Step

1) Capture the code from $1000-$15FF.

```
CALL-151
9600<C600.C6F7M
96F8:A9 59 8D 80 08 A9 FF 8D 81 08
  4C 01 08
C600G
1008:EA EA EA
136C:AA
```

Insert a normal disk.

```
6(CTRL-P)
BSAVE CODE,A$1000,L$600
```

2) Type in the Super IOB controller at the end of this article and run it. Remember to format the disk with a volume of 1. The program checks the volume to differentiate between the program disk and the data disk.

3) Have fun!

## controller

```
1000 REM SCIENCE TOOL KIT CONTROLLER
1010 TK = 0 :ST = 0 :CD = WR : RESTORE : GOSUB
     170 : GOSUB 490
1020 GOSUB 430 : GOSUB 100 :ST = ST + 1 : IF ST <
     4 THEN 1020
1030 GOSUB 1120
1040 GOSUB 230 : GOSUB 490 :ST = 0
1050 GOSUB 430 : GOSUB 100 :ST = ST + 1 : IF ST <
     10 THEN 1050
1060 TK = 2 :LT = 32 :ST = 15 :LS = 15 :CD = WR :FAST
     = 1
1070 RESTORE : GOSUB 170 : GOSUB 490 : GOSUB 610
1080 GOSUB 230 : GOSUB 490 :T1 = TK :TK = PEEK (TRK
     ) - 1 : RESTORE : GOSUB 310 :TK = T1 : GOSUB
     610
1090 IF PEEK (TRK ) = LT THEN 1110
1100 TK = PEEK (TRK ) :ST = PEEK (SCT ) : GOTO 1070
1110 HOME : PRINT "COPY^ DONE!" : END
1120 A$ = "275E:7^ 5^ 3^ 1^ E^ C^ FF^ N ^
     2846:A2^ 0^ BD^ 0^ 11^ 9D^ 0^ 15^ E8^ D0^
     F7^ CE^ 4A^ 9^ CE^ 4D^ 9^ AD^ 4A^ 9^ C9^
     B^ D0^ E8^ 60^ N^ D9C6
1130 FOR A = 1 TO LEN (A$ ) : POKE 511 + A , ASC (
     MID$ (A$ ,A ,1 ) ) + 128
1140 NEXT : POKE 72 ,0 : CALL - 144 : RETURN
5000 DATA 222 ,187 ,222 ,170
5010 DATA 2^ CHANGES
5020 DATA 8 ,8 ,108 ,170
5030 DATA 4 ,10 ,159 ,170
10010 PRINT CHR$ (13 ) CHR$ (4 ) "BLOAD CODE,
     A$2B00"
```

## controller checksums

```
1000  - $356B      1100  - $EAA4
1010  - $98C6      1110  - $AB1F
1020  - $D243      1120  - $53C6
1030  - $590E      1130  - $CD1A
1040  - $D3D6      1140  - $0B37
1050  - $E45A      5000  - $571D
1060  - $DF14      5010  - $6B26
1070  - $0BC5      5020  - $6C93
1080  - $B0CC      5030  - $8BDE
1090  - $5D4F      10010 - $9FCF
```

★★★★★★★

**IMPORTANT**

★★★★★★★

# NOTICE

**For readers of**

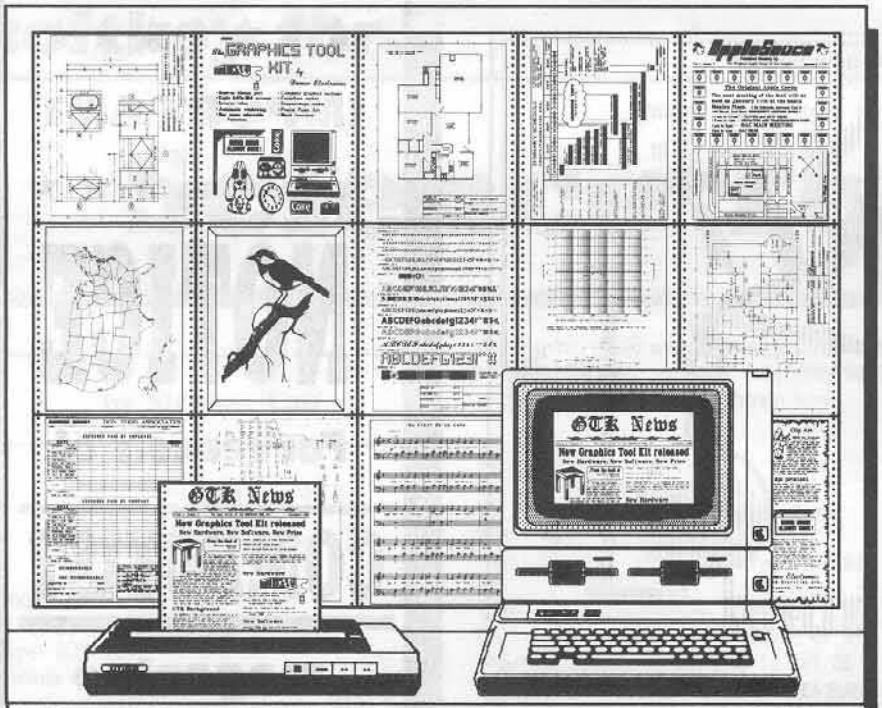
# COMPUTIST

**magazine**

**This issue (46) of COMPUTIST is late because it is now being edited and published by an ALL VOLUNTEER staff of dedicated users.**

**COMPUTIST magazine itself will be undergoing some changes, too. And the next issue (47) will reflect these improved editorial and layout procedures!**

**(ed.)**




MACINTOSH GRAPHICS FOR THE APPLE II

The **Graphics Tool Kit**

*Demco Electronics*

10516 Grevillea Ave.
Inglewood, Ca. 90304
(213) 677-0801

All artwork for this ad was produced with the GRAPHICS TOOL KIT

**APPLE COMPATIBLES LOWEST PRICES ANYWHERE!**

| | |
|---|---|
| 64/80 Column BD (IIe) | $ 39 |
| 256K/80 Column BD - w/Software (IIe) | $109 |
| 80 Column BD - Videx Compatible (II+) | $ 49 |
| Z80 CP/M BD (II+/IIe) | $ 38 |
| 16K Ram BD - w/Cable (II+) | $ 35 |
| 128K Ram BD (II+/IIe) | $ 89 |
| Graphic Parallel BD - w/Cable (II+/IIe) | $ 45 |
| Super Serial BRd - w/cable | $ 49 |
| GS Super Cooler Fan | $ 25 |
| Disk Drive HH (Specify II+/IIe or IIc) | $129 |
| Numeric Key Pad-16 Keys (IIe) | $ 35 |
| Coolin Fan w/Surge Protector (II+/IIe) | $ 28 |
| Joy Stick (Specify II+/IIe or IIc) | $ 15 |
| Joystick w/Large Fire Button | $ 25 |
| Mini Vacumn 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

**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 Llylgamyn | Mask Of The Sun | Minit Man | Mouskattack | Music Construction Set | Oil's Well | Pandora's Box | Robotron | Sammy Lightfoot | Screenwriter II v2.2 | Sensible Speller 4.0, 4.0c, 4.1c | the Spy Strikes Back | Time Zone v1.1 | Visible Computer: 6502 | Visidex | Visiterm | Zaxxon | Hayden Software | Sierra Online Software | **PLUS** the complete listing of the ultimate cracking program...Super IOB 1.5 | **and more!**

**Volume III:** Compiled from issues 11-15

**contains softkeys for:** Alien Addition | Alien Munchies | Alligator Mix | Computer Preparation SAT | Cut And Paste | Demolition Division | DLM (Developement 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 | Csaver | 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 _____

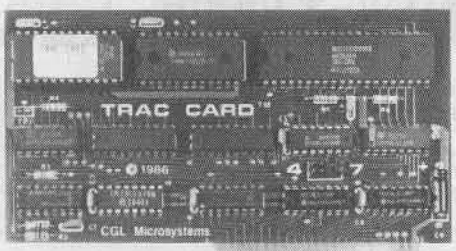
VISA MasterCard _____-_____-_____-_____ Exp. _____

Signature _____ CP46

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**

## 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

## 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
DuoDdisk, 5 1/4" Unidisk and IIc Owners Please Write

Apple is a registered trademark of Apple Computer Inc.

Personal checks, M.O., Visa and Mastercard
Phone 913 676-7242

**Midwest MM Microsystems**

10308 Metcalf, Suite 355
Overland Park, KS. 66212


# the COMPUTIST shopper

| | | | | | |
|---|---|---|---|---|---|
| Animate | $43.00 | Lode Runner | $22.00 | Sargon III | $25.00 |
| Award Maker | $23.75 | **Magic Mindow II** | **$88.00** | Silent Service | $23.00 |
| Bank Street Writer + (128K) | $46.75 | **Magic Window //e** | **$88.00** | Summer Games II | $25.00 |
| Black Cauldron | $25.00 | Math Blaster | $27.00 | **Super Macroworks** | **$29.50** |
| Certificate Maker | $28.00 | Math Rabbit | $25.00 | Where in the USA Carmen San Diego | $26.50 |
| **Copy II+** | **$23.00** | 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 | **Word Perfect w/ Spelling Checker** | **$95.00** |
| **GPLE** | **$29.50** | Newsroom | $35.00 | WordPerfect (GS) | $95.00 |
| Gamemaker | $32.00 | Paintworks + (GS) | $52.00 | World Games | $25.00 |
| Hacker II | $25.00 | **Printshop** | **$32.00** | Writer Rabbit | $25.00 |
| Hacker II (GS) | $30.00 | **Printshop Companion** | **$26.00** | 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 Llylgamyn | $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 _____

VISA _____ - _____ - _____ - _____ Exp. _____

Signature _____ CP46

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

| Issue | Mag $4.75 | Disk $9.95 | Both $12.95 |
|---|---|---|---|
| 45 | ☐ | ☐ | ☐ |
| 44 | ☐ | ☐ | ☐ |
| 43 | ☐ | ☐ | ☐ |
| 42 | ☐ | ☐ | ☐ |
| 41 | ☐ | ☐ | ☐ |
| 40 | ☐ | ☐ | ☐ |
| 39 | ☐ | ☐ | ☐ |
| 38 | ☐ | ☐ | ☐ |
| 37 | ☐ | ☐ | ☐ |
| 36 | ☐ | ☐ | ☐ |
| 35 | ☐ | ☐ | ☐ |
| 34 | ☐ | ☐ | ☐ |
| 33 | ☐ | ☐ | ☐ |
| 32 | ☐ | ☐ | ☐ |
| 31 | ☐ | ☐ | ☐ |
| 30 | ☐ | ☐ | ☐ |
| 29 | ☐ | ☐ | ☐ |
| 28 | ☐ | ☐ | ☐ |
| 27 | ☐ | ☐ | ☐ |
| 26 | ☐ | ☐ | ☐ |
| 25 | ☐ | ☐ | ☐ |
| 24 | ☐ | ☐ | ☐ |
| ☆ 23 | ☐ | ☐ | ☐ |
| 22 | ☐ | ☐ | ☐ |
| 20 | ☐ | ☐ | ☐ |
| 19 | ☐ | ☐ | ☐ |
| 16 | ☐ | ☐ | ☐ |
| 8 | • | ☐ | • |
| ☆ 7 | ☐ | ☐ | ☐ |
| Core 2 | ☐ | ☐ | ☐ |
| 1 | ☐ | ☐ | ☐ |
| Core 1 | ☐ | ☐ | ☐ |
| Core 3 | ☐ | ☐ | ☐ |
| Computing 3 | ☐ | • | • |
| Best of Hardcore Computing | • | ☐ | • |
| Core Special $10.00 (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

# COMPUTIST back issues and library disks

**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 44.

## •••••• Back Issues and Library Disk Rates ••••••

**Back Issues:** US, Canada and Mexico -**$4.75** each. All other Foreign -**$8.75** each.
**Library Disks:** US, Canada, Mexico -**$9.95** each. All other Foreign -**$11.94** each.

**"Both" disk and magazine special COMBO offer:**
US, Canada & Mexico -**$12.95** each combination. other Foreign -**$18.95** each combination.

## 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 • Zyphyr Wars • Vector Graphics • Review of Bit Copiers • Boot Code Tracing • Softkey IOB • Interview with 'Mike' Markkula • .....

**Please send the back issues and library disks indicated.** *Send check/money order to:*

Name _____ ID# _____
Address _____
City _____ State _____ Zip _____
Country _____ Phone _____
VISA MasterCard _____-_____-_____-_____ Exp._____
Signature _____ CP46

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

**(206) 474-5750**

US funds drawn on US banks

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.

Got a problem?

## You can't find COMPUTIST in any computer-software store?

**Of course you can't. That's because COMPUTIST is an UNDERGROUND computer-user's magazine.**

# SUBSCRIBE!

**Time to Renew?** Check your mailing label to see if you need to renew your subscription.

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

**Combo Subs...** You can upgrade your current subscription to a magazine-disk combination by sending $5.50 ($6.50 Foreign) per each remaining issue.

**Use this order form to begin or renew your subcription.**

*Yes, I want an annual subscription to COMPUTIST.*
I am... ☐ A new Subscriber.
☐ Renewing my current subscription
☐ Changing my address (I include latest label).

☐ US — $32
☐ US-Canada-Mexico First Class — $45
☐ US-Canada-Mexico First Class disk-magazine Combo — $100
☐ All other Foreign — $75
☐ All other Foreign disk-magazine Combo — $140

Name _____ ID# _____
Address _____
City _____ State _____ Zip _____
Country _____ Phone _____
VISA MasterCard _____-_____-_____-_____ Exp._____
Signature _____ CP46

*US funds drawn on US banks.* Please allow 4 to 8 weeks for subscription to begin.
Send orders to: **COMPUTIST PO Box 110846-T Tacoma, WA 98411**

# Back Issues

**US/Canada/Mexico - $4.75 ea  Foreign - $8.75 ea**

**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 /// • *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 Apshai Trilogy • Troll Associates programs • Spell It • Regatta • Cdex Training programs • Think Fast • *Features* • How to Write-Protect your Slot Zero • Capturing Locksmith 6.0 Fast Copy • Revisiting DOS to ProDOS and Back • *Core* • Computer Eyes / 2: a Review • *APTs* • Sword of Kadash & Rescue Raiders • Ultimaker IV.....

**42** *Softkeys* • Light Simulator • *Readers' Softkeys* • Beach-Head • Monty Plays Scrabble • Racter • Winnie the Pooh • Infocom Stuff, Kabul Spy, Prisoner II • Wizardry 1 & 2 • Lucifer's Realm • The PFS Series • Dollars and Sense • Strip Poker • Coveted Mirror • Wizard's Crown • The Swordthrust Series • Axis Assassin • Manuscript Manager • The Crown of Arthain • Address Book • Decimals 3.0 • Dragonfire • *Features* • Auto Duel Editor • Wizard's Crown Editor • Questron Mapper • *Core* • The Games of 1986 in Review • *Adventure Tips* • Ultima IV.....

**41** *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 ][ • *Readers' Softkeys* • Discovery! • M-ss-ng L-nks series • Donald Ducks's Playground • Mastering the SAT • Copy ][ Plus 4.4C • Master of the Lamps • One on One • Bridge Baron • A.E. • Great American Cross-Country Road Race • Computer Preparation for the SAT • Castle Wolfenstein • Luscher Profile • Skyfox • Silent Service • Echo Plus • Swashbuckler • Randamn • *Features* • Electronic Disk Drive Swapper • Abusing the Epilogues • Print Shop Companion's Driver Game • *Core* • Keyboard Repair • Fixing the Applesoft Sample Disk • *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 //e • Looking into Flight Simulator's DOS • *Core* • Appavarex • Installing a RAM disk into DOS 3.3.....

**37** *Softkeys* • Under Fire • Pegasus ][ • Take 1 (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 ][ 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 installng a CPS Clock Driver.....

**34** *Softkeys* • Crisis Mountain • Terripin Logo • Apple Logo II • Fishies 1.0 • SpellWorks • Gumball • *Readers' Softkeys* • Rescue at Rigel • Crazey 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 • Kracking Vol II • Moebius • Mouse Budget, Mouse Word & Mouse Desk • Adventure Construction Set • *Feature* • Using Data Disks With Microzines • *Core* • Super IOB v1.5 a Reprint.....

**31** *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.....

**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 • Apventure to Atlantis • *Feature* • Capturing the Hidden Archon Editor • *Core* • Fingerprint Plus: A Review • Beneath Beyond Castle Wolfenstein (part 2).....

**27** *Softkeys* • Microzines 1-5 • Microzines 7-9 | Microzines (alternate method) • Phi Beta Filer • Sword of Kadash • *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 1 • 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 • Homeword Speller • *Feature* • Adding IF THEN ELSE To Applesoft • *Core* • DOS To ProDOS And Back.....

**24** *Softkeys* • Electronic Arts software • Grolier software • Xyphus • F-15 Strike Eagle • Injured Engine • *Readers' Softkeys* • Mr. Robot And His Robot Factory • Applecillin II • Alphabet Zoo • Fathoms 40 • Story Maker • Early Games Matchmaker • Robots Of Dawn • *Feature* • Essential Data Duplicator copy parms • *Core* • Direct Sector Access From DOS....

**23** *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 ][ 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 1.....

**1** *Softkeys* • Data Reporter • Multiplan • Zork • *Features* • PARMS for Copy II Plus • No More Bugs • APT's for Choplifter & Cannonball Blitz • *'Copycard' Reviews* • Replay • Crackshot • Snapshot • Wildcard

# 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 Csaver program from COMPUTIST No. 13, and a Menu Hello Program that lists the available controllers and, when you select one, automatically installs it in Super IOB and RUNs the resulting program.*

► A reprint of **Disk Inspection and the Use of Super IOB**, from COMPUTIST No. 17. This article explains how to write your own Super IOB controllers.

► **COMPUTIST No. 32**, which contains an extensive article detailing the hows and whys of Super IOB v1.5 and at least 5 articles using the new Super IOB program.

- Several of the controllers deprotect the software completely with no further steps. This means that some programs are only minutes away from deprotection (with virtually no typing).
- The issue of COMPUTIST in which each controller appeared is indicated in case further steps are required to deprotect a particular program.**

### The SUPER IOB Collection

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

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

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

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

☐ **Yes, please send me The Super IOB Collection**

Includes both disks with Super IOB version 1.5, COMPUTIST#32, PLUS a reprint of "Disk Inspection and the Use of Super IOB".

☐ **US/Canada/Mexico for $16 00**
☐ **Other Foreign for $20.00**

**Send to:** **Super IOB Collection**
**PO Box 110846-T**
**Tacoma, WA 98411**
**(206) 474-5750**

Name _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

VISA MasterCard _____-_____-_____-_____ Exp._____

Signature _____ CP46

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**

*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.