

# COMPUTIST

Issue 55

May 1988

USA \$3.75

Canada & Mexico \$7.00

All Others \$8.75

## 🍏 RDEX *Features:*

■ **A Utility to Save the Lower 8 Pages of Memory**—By Mark Harris

■ **Bard's Tale Effects Locator**—By George Bigelow

■ **How to Capture *Phantasia* Screen Maps**—By George Bigelow

■ ***Alternate Reality* Character Editor**—By Michael Horton

■ **Updating the ProDOS Block Editor**—By Rene Gaudet

■ **Loading *Flashcalc* onto your RAMcard**—By John Pierce

■ **A Copy-protection Scheme for ProDOS**—By Stanley Planton

■ **The Product Monitor**—By Jeff Hurlburt

■ ***Autoduel* Car Editor**—By Marc Batchelor

## 🍏 RDEX *Softkeys:*

Alphabet Sequencing 🍏 Animal Alphabets and Other Things 🍏 Arctic Antics 🍏 The Boars' Store 🍏 The Boars Tell Time  
🍏 Career Focus 🍏 Castle Wolfenstein 🍏 Charlie Brown's 123's 🍏 Charlie Brown's ABC's 🍏 City Country Opposites 🍏 Coveted  
Mirror 🍏 Create With Garfield 🍏 Crypt of Medea 🍏 Customized Alphabet Drill 🍏 Customized Flash Spelling 🍏 Dig Dug  
🍏 Digital Paintbrush System 🍏 Estimation 🍏 Fay: Word Hunter 🍏 Fix It 🍏 Focusing on Language Arts 🍏 Fundamental  
Capitalization 🍏 Fundamental Punctuation Practice 🍏 Fundamental Spelling Words in Context 🍏 The Hobbit 🍏 Homonyms  
in Context 🍏 Individualized Study Master 🍏 Inside Outside Shapes 🍏 Inside Outside Opposites 🍏 Leisure Suit Larry in the  
Land of the Lounge Lizards 🍏 Master Diagnostics II & II+ 🍏 Mastertype v2.1 🍏 Mathematics Series 🍏 Mr. and Mrs. Potato  
Head 🍏 Paper Models - The Christmas Kit 🍏 Peanuts Math Matcher 🍏 Peanuts Maze Marathon 🍏 Peanuts Picture Puzzlers  
🍏 Perry Mason: The Case of the Mandarin Murder 🍏 Railroad Works 🍏 Random House Library 🍏 Management Programs  
🍏 Rocky's Boots v.4 🍏 Sensible Speller 🍏 Snoopy's Reading Machine 🍏 Snoopy's Skywriter Scrambler 🍏 Snoopy to the  
Rescue 🍏 Snoopy Writer 🍏 Spelling Demons 🍏 Stock Market Simulation 🍏 Story Builder 🍏 Story Starter 🍏 Studio II 🍏 Test  
Maker 🍏 Think Quick v1.0 🍏 Tournament Bridge 🍏 Tutorial Comprehension 🍏 Typing is a Ball, Charlie Brown 🍏 Under  
Fire 🍏 Word Blaster 🍏 Word Count 🍏 Word Mount 🍏 Your Personal Net Worth

**IBM RDEX:** Feature: ● Flight Simulator RGB Modifications Softkey: ■ Samna Word II v1.1

COMPUTIST  
PO Box 110846-T  
Tacoma, WA 98411

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

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

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

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

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

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

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

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

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

*United States Code title 17, §117*

## COMPUTIST

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

● Address all advertising inquiries to:  
COMPUTIST; Advertising Department; PO  
Box 110816; Tacoma, WA 98411

● Mail all RDEX letters to:  
COMPUTIST  
Apple-RDEX or IBM-RDEX  
PO Box 110846-K  
Tacoma, WA 98411

● COMPUTIST does NOT purchase editorial material. The entire editorial content consists of information submitted to COMPUTIST for publication in the shared interests of all COMPUTISTS.

● Unsolicited material (manuscripts, letters to the editor, softkeys, A.P.T.s, playing tips, questions, etc., are assumed to be submitted as letters-to-the-RDEX-editor for publication with all and exclusive rights belonging to COMPUTIST.

● Entire contents copyright 1988 by SoftKey Publishing. All rights reserved. Copying done for other than personal or internal reference (without express written permission from the publisher) is prohibited.

● The volunteer and paid editorial staff assume no liability or responsibility for the products advertised in the magazine. Any opinions expressed by the authors are not necessarily those of COMPUTIST magazine, its staff or SoftKey Publishing.

● Apple® is a trademark of Apple Computers. IBM® is the IBM trademark.

**SUBSCRIPTIONS:** Rates (for 12 issues):  
U.S.—\$32 U.S. 1st Class—\$45  
Canada/Mex.—\$45 Other Foreign—\$75

● Send subscription inquiries to:  
COMPUTIST; Subscription Department; PO  
Box 110846-T; Tacoma, WA 98411

● For 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.



**Attention Apple-users:  
Why type those  
l-o-n-g listings when  
you can just Upgrade  
your present sub. to a  
Mag/5 1/4" Disk Combo  
subscription**

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

**Time to renew?**

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

**Moving soon?**

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

*Subscribe*

*to*


*Computist*

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

- I am...  A new subscriber  
 Renewing my current subscription  
 Changing my address (please include last mail label)
- I have...  an Apple IIe, IIgs or IIc computer  
 an IBM computer

- Apple-IBM subscription...
- U.S. - \$32
  - U.S./Canada/Mexico First Class - \$45
  - All other Foreign - \$75

- Apple mag-disk combination subscription...
- U.S./Canada/Mexico First Class plus Library Disk - \$100
  - All other Foreign plus Library Disk - \$140

Name \_\_\_\_\_ ID# \_\_\_\_\_  
 Address \_\_\_\_\_  
 City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_  
 Country \_\_\_\_\_ Phone \_\_\_\_\_  
 \_\_\_\_\_ Exp. \_\_\_\_\_  
 Signature \_\_\_\_\_ CP55

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

## APPLE RDEX Contributors

J. Aufderheide.....	18
Ron Balewski.....	35
Marc Batchelor.....	27
George Bigelow.....	9
Marshall P. Brown.....	33
Steve Cook.....	37
Dorothy Dow.....	7
Rene Gaudet.....	14
David Goforth.....	34
Kim Griffith.....	22
Mark Harris.....	7
Jim Hart.....	24
Michael A. Horton.....	11
Jeff Hurlburt.....	25
Francisco-Angel Garcia Moruja.....	31
Jack R. Nissel.....	19
John R. Pierce.....	20
Stanley Planton.....	20
Jerry Stevens.....	13
Charles Taylor.....	13
J.D. Tischer.....	37
R.D. West.....	10
Greiner Wolfgang.....	32
Zorro.....	19

## IBM RDEX Contributors

Mike Basford.....	44
-------------------	----

# Notes 'n things.....

## ■ Writing to authors

Some readers missed my note on how to send letters to authors. If the author puts a line in their article that states "Please print my address" then we will include the address when we print their contribution. However, if they don't clearly state that it is alright to print their address then we don't. Sometimes, we have been known to delete addresses even when they want them printed. Sorry, force of habit.

Even if there is no address printed, you can still write to an author if they have given us their address. We have a back-up method. (Hackers always have a back-up.)

1. Write your letter and enclose it in an envelope with the author's name on the front and the correct postage.
2. Put that letter into another envelope and sent it to us.
3. When we receive your letter, we will complete the forwarding address and drop it in the outgoing mail.
4. The author may then contact you directly.

**Note:** Some authors are very secretive and don't even let us know where they are. Also, if you are considering having us print your address, use a Post Office box instead of your street address. If you read the news papers, you'll know that there are a lot of weird people out there. You may not want them to know where you live.

# BUGS

## COMPUTIST #50, page 37

Capturing Locksmith Fast Copy for IIGs. Where it says "SAVE FC, A\$2000, L\$18FD", It should say "BSAVE FC, A\$2002, L\$18FD".



### **Softkeys:**

Alphabet Sequencing.....21	Peanuts Math Matcher.....21
Animal Alphabets and Other Things.....21	Peanuts Maze Marathon.....21
Arctic Antics.....35	Peanuts Picture Puzzlers.....21
The Boars' Store.....21	Perry Mason: The Case of the Mandarin Murder.....19
The Boars Tell Time.....21	Railroad Works.....37
Career Focus.....21	Random House Library Management Programs.....21
Castle Wolfenstein.....19	Rocky's Boots v.4.....24
Charlie Brown's 123's.....21	Sensible Speller.....8
Charlie Brown's ABC's.....8, 21	Snoopy's Reading Machine.....21
City Country Opposites.....21	Snoopy's Skywriter Scrambler.....21
Coveted Mirror.....24	Snoopy to the Rescue.....21
Create With Garfield.....8, 13	Snoopy Writer.....21
Crypt of Medea.....32	Spelling Demons.....21
Customized Alphabet Drill.....21	Stock Market Simulation.....21
Customized Flash Spelling.....21	Story Builder.....21
Dig Dug.....7	Story Starter.....21
Digital Paintbrush System.....33	Studio II.....23
Estimation.....18	Test Maker.....34
Fay: Word Hunter.....10	Think Quick v1.0.....8
Fix It.....21	Tournament Bridge.....21
Focusing on Language Arts.....21	Tutorial Comprehension.....21
Fundamental Capitalization.....21	Typing is a Ball, Charlie Brown.....21
Fundamental Punctuation Practice.....21	Under Fire.....24
Fundamental Spelling Words in Context.....21	Word Blaster.....21
The Hobbit.....19	Word Count.....21
Homonyms in Context.....21	Word Mount.....21
Individualized Study Master.....21	Your Personal Net Worth.....31
Inside Outside Shapes.....21	
Inside Outside Opposites.....21	
Leisure Suit Larry in the Land of the Lounge Lizards.....19	
Master Diagnostics II & II+.....22	
Mastertype v2.1.....13	
Mathematics Series.....21	
Mr. and Mrs. Potato Head.....21	
Paper Models - The Christmas Kit.....36	

### **Playing Tips:**

Under Fire.....23

### **IBM Softkeys:**

Samna Word II v1.1.....44

### **IBM Notes:**

Flight Simulator RGB Modifications 44

Editorial.....4	4
Bugs.....4	4
Most Wanted Softkeys.....37	37

### **Features:**

Memory Save Utility.....8	8
Mark Harris shows how to save the Lower 8 Pages of Memory.	
Bard's Tale Effects Locator.....9	9
George Bigelow's supplement to Bard's Tale Mapper that tells you what the effect numbers really are.	
Phantasia Map Capture.....10	10
George Bigelow's easy method for capturing the screen images from Phantasia	
Alternate Reality Character Editor....11	11
Michael Horton has made character editing easier for Alternate Reality enthusiasts.	
Updating the...	
ProDOS Block Editor.....14	14
Rene Gaudet tackles the chore of adding a search command and while he's at it, extends the block read capability to handle 3.5" disks.	
Loading your RAMcard with...	
Flashcalc.....20	20
John Pierce shares his step-by-step procedure for installing Flashcalc on an 800K RAMdisk.	
A Copy-protection Scheme for...	
ProDOS.....20	20
Stanley Planton has devised a method to copy protect ProDOS files using EDD 4.N.	
The Product Monitor.....25	25
Jeff Hurlburt reports on the winners and losers in the ongoing joystick battle.	
Fix Up Your Car with...	
Autoduel Car Editor.....27	27
Marc Batchelor's passion for AutoDuel has led to him write a user-friendly car editor.	





# Readers Data

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

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

## What is a softkey, anyway?

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

## Commands and control keys

In any article appearing in COMPUTIST, commands which a reader is required to perform are set apart by being in boldface and on a separate line. The **RETURN** key must be pressed at the end of every such command unless otherwise specified. Control characters are specially boxed. An example of both is:

**6** **OP**

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

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

## Special requirements

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

### ■ Requirements:

### Software recommendations

✓ Applesoft program editor such as *Global Program Line Editor (GPL)*.

✓ Sector-editor such as *DiskEdit* (in the *COMPUTIST Starter Kit*) or *ZAP* from *Bag of Tricks*.

✓ Disk-search utility such as *The Inspector*, the *CIA* or the *Core Disk Searcher* (in the *COMPUTIST Starter Kit*).

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

✓ Bit-copy program such as *Copy II Plus*, *Locksmith* or *Essential Data Duplicator (EDD)*.

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

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

## Super IOB and Controllers

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

## Reset into the Monitor

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

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

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

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

## Recommended literature:

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

## Keying in Applesoft programs:

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

**10HOME:REMCLEAR SCREEN**

The LIST will look like:

10 HOME : REM CLEAR SCREEN

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

## Keying In Hexdumps

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

**CALL -151**

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

When finished, return to BASIC with:

**3DOG**

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

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

## Computing checksums

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

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

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

## CHECKSOFT instructions:

**LOAD filename**

**BRUN CHECKSOFT**

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

## CHECKBIN instructions:

**CALL -151**

**BLOAD filename**

Install *CHECKBIN* at an out of the way place

**BRUN CHECKBIN, A\$6000**

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

**SSS.EEE Y**

Correct the lines at which the checksums differ.  
..... RDXed



# EXchange



etc... who want all their software backed up and COPYA-able



when writing a letter to...

## Apple RDEXed

### RDEX stands for: Reader's Data EXchange

That means that when you send in articles, softkeys, APTs, etc., you are submitting them for FREE publication in this magazine. RDEX does NOT purchase submissions nor do we verify data submitted by readers. We will print it and it is the responsibility of the readers to send in responses when anything is wrong.

• Remember that your letters or parts of them may be used in RDEX even if not addressed to the Apple-RDEX editor. Correspondence that gets published may be edited for clarity, grammar and space requirements.

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

### Send your articles and letters on DOS 3.3 standard text files

When we get your letter-article in a standard DOS3.3 text file, it is immediately uploaded into the most current RDEX file. Conventional letters must be typed in by us...when we have the time.

• Address your letters, articles, to:

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

Dorothy Dow

My computer is an Apple IIgs. I am both a new subscriber to your magazine and new to using a computer, as such, I have several questions.

The Apple IIgs uses ProDOS. How do I get DOS 3.3 or do I need it. Your instructions on page 6, COMPUTIST #51, do not mention the IIgs at all.

Where do I find GPLE. Do I need a CD ROM like the IIc and IIe?

Please update your directions to include the IIgs so beginners like me can figure out where to start.

*You've asked some very good questions. Unfortunately, I don't have the answers. It's sad, but we don't have a IIgs and we may not be getting one soon. Anything I say about the IIgs is hearsay, guesswork or out of a book. Perhaps the best thing would be to ask our readers to answer your questions. How about it readers, has anyone had their IIgs long enough to hazard updating the Data page for us and answering Dorothy's questions. RDEXed*

Mark Harris

Softkey for...

### Dig Dug DataSoft

#### Requirements

- An initialized slave disk (use a fast DOS)
- A way to force entry into the monitor.
- A way to save the lower 8 pages of memory. (Senior Prom, Modified Roms etc.)

Dig Dug is a fun arcade game that has been around for quite some time. I do not recall if there is an earlier softkey for this program, so I thought I would pass mine along.

Follow these steps to deprotect Dig Dug into binary files.

1 Boot your original disk and enter the monitor when the game starts to play.

2 Relocate some code:  
4000<8000.BF00M

3 Boot your slave disk and save the relocated code.

#### BSAVE DG.OBJ, A\$4000, L\$4000

4 Boot your original again and enter the monitor while the title screen is still being displayed, then boot your slave disk and save the title page.

#### BSAVE DG.LOGO, A\$2000, L\$2000

5 Boot the original once again and enter the monitor when the game is being played.

6 Move the lower 8 pages of memory to a safe location with whatever method you have.

7 Boot your slave disk and save the 8 pages that you moved. The following assumes that you moved it to \$2000.

#### BSAVE DG.P0-P7, A\$2000, L\$800

8 Enter the monitor, type the following hex dump and save it:

#### DG.MOVER

```
0900:A0 00 A2 00 BD 00 40 9D      $22A0
0908:00 80 E8 D0 F7 EE 06 09      $F7B4
0910:EE 09 09 C8 C0 40 D0 EA      $3DBE
0918:A0 00 A2 00 BD 00 10 9D      $8777
0920:00 00 E8 D0 F7 EE 1E 09      $FE0B
0928:EE 21 09 C8 C0 08 D0 EA      $4451
0930:A0 00 A2 00 BD 00 20 9D      $0618
0938:00 40 E8 D0 F7 EE 36 09      $3BDC
0940:EE 39 09 C8 C0 20 D0 EA      $B13E
0948:4C 00 80                      $7EF7
```

#### 3D0G

#### BSAVE DG.MOVER,A\$900,L\$4B

9 Enter this Applesoft hello program and save it:

#### FP

```
10 TEXT : HOME : VTAB 10: HTAB 11: PRINT
"ONE MOMENT PLEASE" : PRINT CHR$
(4) "BLOAD DG.OBJ,A$4000" : PRINT
CHR$ (4) "BLOAD DG.P0-P7,A$1000" :
PRINT CHR$ (4) "BLOAD DG.LOGO,
A$2000"
```

```
20 POKE - 16304,0: POKE - 16302,0: POKE -
16300,0: POKE - 16297,0: PRINT CHR$
(4) "BRUN DG.MOVER"
```

#### SAVE HELLO

That's it. If it doesn't work, chances are you either missed one of the above steps or you did not save the lower 8 pages unaltered. Try again.

Softkey for...

## Think Quick V1.0

Learning Company

### ■ Requirements

- 2 ProDOS formatted disks
- A Copy program
- A Block Editor
- Think Tank V1.0*
- Think Tank* data disk

This softkey requires the use of a block editor. This is much like the sector editor found on *Copy II Plus*, only the data storage format is a bit different. I used a new program available on bulletin board services called *Block Work*. This is a very good program and is extremely flexible. It's marketed as shareware and the author is asking \$15.00 for the program. I think it is well worth the money. So make your job easier and use this type of program.

*Think Quick* is a good logic oriented game aimed at the elementary aged person. It is relatively easy to use and fairly interesting (from an adults point of view). *Think Quick* is ProDOS based, which is good news for disk access speed.

**1** Copy both sides of the program and data disks.

**2** Using a block editor, search both sides of the program disk for 20 9E 63 and replace it with EA EA EA. On my disk these bytes were found on Block \$15, starting at byte \$D1.

**3** Now search for 20 12 17 and replace it with EA EA EA. On my disk these bytes were found on Block \$54, starting at byte \$08.

You don't need to worry about the data disk, it is not protected.

Softkey for...

## Charlie Brown's ABC's

Random House Software

### ■ Requirements

- Initialized disk
- COPYA or equivalent

This is a great program for preschool age people. I had a good time going through the alphabet to see all the wonderful screens. I would highly recommend this program.

To get a working copy follow these directions.

**1** Boot your DOS 3.3 system disk.

**2** Tell DOS to ignore checksum and epilogues and run COPYA.

## POKE 47426,24 RUN COPYA

**3** Run any copy program that will allow you to copy a fast DOS, such as *Copy II Plus*, etc. Make sure that your boot program is named HELLO.

**4** Repeat the same procedure on the back side of your disk and you will be done. Have fun.

Softkey for...

## Create With Garfield

DLM

### ■ Requirements

- Blank disk
- Copy program
- Sector Editor
- Original disk

On page 7 of COMPUTIST #44, Robert Brown sent in a softkey for *Create with Garfield*. Unfortunately, it was a bit confusing and, for my version at least, incomplete. I had to make a few more edits to my version. I do not have the original bytes to search for, and I apologize. They will have to be absolute.

Make the following sector edits:

Original edits from COMPUTIST #44.

Trk	Sct	Byte(s)	From	To
\$1D	\$02	\$18	?	A2 00 60
\$1F	\$00	\$F0	?	A2 00 60
\$1F	\$06	\$82	?	A2 00 60
\$22	\$04	\$49	?	A2 00 60

My new Edits.

Trk	Sct	Byte(s)	From	To
\$1F	\$06	\$B2	?	A2 00 60
\$1B	\$0E	\$DD	?	A2 00 60

Make sure you write these edits back to your disk.

Softkey for...

## Sensible Speller

Sensible Software

### ■ Requirements

- ProDOS formatted disk
- Block Editor, like *Block Work*
- Copy Program, like *Block Work*
- Original disk

*Sensible Speller* is a ProDOS based spelling checker, which seems to be very effective. I have not used it much, but it is easy to operate. The copy protection scheme is similar to the

other pieces of software that Sensible Software market. I found that after the program checked for an original disk, it checked something in RAM memory and would give you an error if it didn't check out. So I found the routine and NOPed it. I then tried using it from a RAM disk, and found that it checked slot 6 for the original disk. Although the check is satisfied by our first edits, it is still a pain to wait for it. The third set of edits will stop this check from happening, and you'll be presented with the main menu screen almost immediately. Plus you'll be able to run it from a hard disk or RAM disk. Perform the following edits to deprotect this program.

**1** Fix Key disk check so it's always satisfied. Search for this sequence BD 8C C0 10 FB 49 D5 D0 F7. On my version I made these changes:

Block	Byte(s)	From	To
\$51	\$0F6	\$F7	\$56
\$78	\$18E	\$F7	\$56
\$F2	\$072	\$F7	\$56

**2** This will fix the RAM check routine that I came across. Search for this sequence: 20 4A 4A. On my version I made these changes:

Block	Byte(s)	From	To
\$5F	\$1EC	20 4A 4A	EA EA EA
\$85	\$1EC	20 4A 4A	EA EA EA

**3** Disable slot 6 drive startup and check (allows hard disk or RAM disk operation). Search for this sequence: 20 04 4A. On my version I made these changes:

Block	Byte(s)	From	To
\$5F	\$1D8	20 04 4A	EA EA EA
\$85	\$1D8	20 04 4A	EA EA EA

After these edits are made your disk will work properly from any storage device.

## Utility for Saving the Lower 8 Pages of Memory

### ■ Requirements

- Formatted disk, DOS not necessary
- Sector Editor

On occasion, you may need to save the lower 8 pages of memory (\$0000-\$0800) to a binary file. Unfortunately, this area of memory is very volatile. It starts with the Zero page, (\$0000-\$00FF) which contains a lot of highly used pointers and variables. It also contains the microprocessor's stack area (\$0100-\$01FF). It has the keyboard buffer





(\$0200-\$02FF). And lastly, it has the text screen display area (\$0400-\$07FF). This memory area changes constantly, which makes it impossible to save this area to disk using normal methods. To get this area saved to disk, we have to somehow move it to a safe area before any keyboarding is performed. Once this is completed, we can save this area to disk like any other binary file.

The following is a procedure for setting up a DOSless utility that will move this area to a safe place so it can be saved to disk. I made use of the information provided by Adam Levin in *COMPUTIST #44*, page 13 on DOSless utilities.

Follow these directions:

**1** Format a blank disk. It doesn't need DOS on it, but it could.

**2** Pull out your favorite sector editor and read in sector #500 from track #500.

**3** Starting at Byte #500 enter in the following code:

```
01 A2 08 A0 00 B9 00 00
99 00 20 C8 D0 F7 EE 07
08 EE 0A 08 CA D0 EC 8D
E8 C0 4C 69 FF
```

**4** When complete, write it back to sector #500, track #500.

To use this utility all you need to do is start up your protected program. Put this utility in the boot drive and press **RESET**. Assuming that this reboots your machine, you'll see the drive come on then you'll be in the monitor. At this time, the lower 8 pages of memory are sitting at address \$2000-\$27FF. Now boot your DOS disk and save this area like this:

**BSAVE P0:P7,A\$2000,L\$800**

If upon hitting **RESET**, the program does not reboot, or it clears memory and reboots, then this method will not work. But if you can drop into the monitor from within your protected program, you may be able to get most of the important data by booting using the command: **6P**

This will disrupt a few memory locations, but it may still work. Give it a try. Good luck.

George Bigelow

## Bard's Tale Effects Locator:

### A Supplement To Bard's Tale Mapper

#### Requirements

- Bard's Tale* (the first one)
- Bard's Tale Mapper* maps
- Printer nice, but not necessary

After I wrote *Bard's Tale Mapper 1* began exploring the dungeon disk some more and discovered how to tell what all those numbers in the effects map that have a 4 in them do (04,84,C4). Each dungeon map has one sector dedicated to this. Perhaps, in your looking through the disk, you have seen a sector with the name of the dungeon. This sector is the one that tells the program what to do for each 4 on that map.

### How To Read Coordinates

Whenever you ask *Bard's Tale* to locate yourself in a dungeon (there are several ways) it answers with so many rooms North and so many rooms East of the room that you entered the dungeon, no matter which floor you are on. Since dungeons are always entered from the lower left-hand corner, this is the basis for my coordinates; so many rooms North, so many East of the lower left-hand corner of each map. So, if we have a coordinate of 13,7 this means the room that is 13 rooms from the bottom of the map, and 7 rooms from the left side of the map (this is the map you have printed out using *Bard's Tale Mapper*). I find it easiest to mark a piece of paper with numbered lines corresponding to each room and measuring with this.

### How Do I Find Out What Happens At Any Particular Coordinate?

There are two methods which will help you find out what happens at a particular coordinate. One is to use your sector editor, find the sector, and use the following information. The other is to type in the following program, and let it do the work! I recommend the latter.

Let's go through and explain each effect. Coordinates are in pairs, first the distance North from the bottom of the map, then the distance East from the left side of the map. If your sector editor displays eight bytes per line this will be a lot easier (I use *Copy II Plus*). The byte I give will be the byte that this effect's section starts with (each section is 10 bytes long).

First find the sector with the name of the dungeon you are interested in. The dungeon maps start at track 50E, sector 50B through track 506 sector 50C, eight sectors per map. All floors of a dungeon have the same name, so if you are interested in, say, the third level of the sewers, find the third occurrence of the name "SEWERS" going from track 50E sector 50B examining in decreasing track and sector numbers.

\$20: Special Effects. This is my name for this section because it contains many different effects. The first set of coordinates corresponds to the number starting at \$30. For example, in the dungeon SEWERS(1) at room 9,4 is special effect number \$11 (which is a spider statue).

\$40: Spell Removal. As you enter this room any special spells (magic mouth, compass, etc.) will go away. This can be a large category, it continues through \$50.

\$60: Place Change. Entering this room will transport you to the corresponding coordinates starting at \$70. For example, in the TOWER(F) entering room 1,16 will move you to room 1,14 (numbers translated to decimal).

\$80: Direction Shift. Entering this room changes the direction you are facing (note that your compass will not point correctly until you leave this room).

\$90: Smoke in your Eyes. Things will look a lot different from now on no matter where you are until you cast a spell to remove it.

\$A0: Removes Hits: Entering this room hurts! Another large category, this continues through \$B0.

\$C0: This one I don't know! It only occurs in two dungeons, both at coordinates 5B,5F. I can't find anything there and didn't include it in the program.

\$D0: Stasis Chamber. You appear to be going through many doors, but your location does not change. You may even be able to get out, but it takes luck!

\$E0: Inscriptions. In these rooms there is an inscription that corresponds to the order of the inscriptions on the previous two sectors.

\$F0: Special Monsters: Not your run of the mill monsters, but sometimes really interesting(!) ones.

### The Program

If all this previous stuff is not your bag, then *Bard's Tale Effects Locator* is for you! It does all the work except for transferring the information to your map.

Basically, all the program does is read the correct sector into the DOS buffer at \$9600, then read out the numbers found at a certain offset (number from the beginning of the buffer). It reads out everything from then on until it reaches a \$FF (255) or until the number of pairs read reaches either 16 or 32, depending on how large the category is. This information is printed as decimal pairs separated by a comma. Transfer this information on your map and you will be a lot more prepared to survive the dungeon!

### Bard's Tale Effects Locator

```
10 REM BARD'S TALE EFFECTS LOCATOR
20 DIM MS(15):MULT=1:MEM=38400:REM LOCATION
OF BUFFER
30 FOR A=768 TO 775:READ B:POKE A,B:NEXT:REM
RWTS ROUTINE
40 FOR A=0 TO 15:READ MS(A):NEXT:REM
LOCATIONS
50 DATA *169.183.160.232.32.217.3.96
```



```

60 DATA CELLARS(0),SEWERS(1),SEWERS(2),
SEWERS(3),CATACOMBS(4),CATACOMBS(5),
CATACOMBS(6),CASTLE(7),CASTLE(8),
CASTLE(9),TOWER(A),THE TOWER(B),THE
TOWER(C),THE TOWER(D),THE TOWER(E),THE
TOWER(F)
70 REM MENU
80 HOME : PRINT "CHOOSE A LOCATION:" : PRINT
90 FOR A = 0 TO 15: PRINT A + 1 : " SPC(1 + (A <
9))MS(A) : NEXT
100 PRINT "17. END"
110 POKE - 16368,0: REM CLEAR ANY PREVIOUS
CHARACTER
120 PRINT : INPUT "CHOOSE A NUMBER (1-17):"
: C: IF C < 1 OR C > 17 THEN 80
130 IF C = 17 THEN END
140 REM CALCULATE TRACK & SECTOR
150 T = 15 - INT (C / 2 + .5):D = C - 1
160 IF C = 1 THEN S = 11: GOTO 190
170 IF C = 2 THEN S = 3: GOTO 190
180 C = C - 2: GOTO 160
190 POKE 47083,0: POKE 47084,T
200 HOME : VTAB 10: HTAB 5: FLASH: PRINT "INSERT A
BARD'S TALE DUNGEON DISK!" : NORMAL
210 VTAB 15: HTAB 1: PRINT "PRINT THE PAGE?" :
GET Z$: IF Z$ = "Y" THEN PRINT : PRINT CHR$(
4) "PR#1"
220 HOME : PRINT "EFFECTS LOCATIONS FOR" :
MS(D)
230 POKE 47085,S: CALL 768: REM READ CHOSEN
TRACK & SECTOR INTO BUFFER
240 PRINT : PRINT "SPECIAL:" : OFFSET = 32:
GOSUB 410
250 PRINT : PRINT "REMOVES SPELLS:" : OFFSET =
64: MULT = 2: GOSUB 370: MULT = 1
260 PRINT : PRINT "PLACE CHANGE:" : OFFSET = 96:
GOSUB 450
270 PRINT : PRINT "DIRECTION SHIFT:" : OFFSET =
128: GOSUB 370
280 PRINT : PRINT "SMOKE IN EYES:" : OFFSET =
144: GOSUB 370
290 PRINT : PRINT "REMOVES HITS:" : OFFSET =
160: MULT = 2: GOSUB 370: MULT = 1
300 PRINT : PRINT "STASIS CHAMBER:" : OFFSET =
208: GOSUB 370
310 PRINT : PRINT "INSCRIPTIONS:" : OFFSET =
224: GOSUB 370
320 PRINT : PRINT "MONSTERS:" : OFFSET = 240:
GOSUB 370
330 PRINT : PRINT : PRINT : PRINT CHR$(4) "PR#0"
340 VTAB 23: HTAB 1: PRINT "PRESS A KEY TO
RETURN TO MENU" : GET Z$
350 GOTO 80
360 REM SUBROUTINES
370 P = 0
380 LOC = PEEK (MEM + OFFSET + P): IF LOC = 255 OR
P = MULT * 16 THEN RETURN
390 PRINT LOC : " PEEK (MEM + OFFSET + 1 + P) " :
400 P = P + 2: GOTO 380
410 P = 0
420 LOC = PEEK (MEM + OFFSET + P): IF LOC = 255 OR
P = 16 THEN RETURN
430 PRINT LOC : " PEEK (MEM + OFFSET + 1 + P) " :
PEEK (MEM + OFFSET + 16 + P) " :

```

```

440 P = P + 2: GOTO 420
450 P = 0
460 LOC = PEEK (MEM + OFFSET + P): IF LOC = 255 OR
P = 16 THEN RETURN
470 PRINT LOC : " PEEK (MEM + OFFSET + 1 + P) " :
PEEK (MEM + OFFSET + 16 + P) " :
PEEK (MEM + OFFSET + 17 + P) " :
480 P = P + 2: GOTO 460

```

### Checksums

10	- \$BADD	250	- \$125D
20	- \$6E0E	260	- \$EE8F
30	- \$ED83	270	- \$5059
40	- \$DC85	280	- \$97BC
50	- \$E4EF	290	- \$C3ED
60	- \$E39C	300	- \$D15E
70	- \$5E21	310	- \$2359
80	- \$2DB8	320	- \$077A
90	- \$50AD	330	- \$349C
100	- \$85FF	340	- \$F342
110	- \$B977	350	- \$AB5A
120	- \$5B48	360	- \$7A8F
130	- \$A925	370	- \$9542
140	- \$7BC0	380	- \$D335
150	- \$80C2	390	- \$6618
160	- \$EC71	400	- \$A86F
170	- \$9629	410	- \$791C
180	- \$DB23	420	- \$7E7B
190	- \$2941	430	- \$D868
200	- \$2208	440	- \$2688
210	- \$E26E	450	- \$762E
220	- \$46EB	460	- \$B28C
230	- \$CD97	470	- \$1013
240	- \$9638	480	- \$808B

## CAPTURING PHANTASIE MAPS

*Phantasie*, by Strategic Simulations, is a popular game, but like many strategy games is more playable with a map to follow. This article will not create these maps for you, but will allow you to capture them for later reference after you have made the dungeon visible by exploring it. This program will work on the first *Phantasie*, and perhaps the rest. I do not have the rest, so it's iffy!

The game only saves one dungeon status at a time, so if you have thoroughly explored one dungeon, saving another dungeon will wipe out the first one. If you need to return later, you start all over again! It is this characteristic that has made it a necessity to make a map of the dungeons.

The people at Strategic Simulations, Inc. have made it easy for us. When **[RESET]** is pressed during the game any hi-res image is not wiped out. This means that if we create a disk with a hello program that can save a picture,

we can save the *Phantasie* dungeon.

Type in the program "Phantasie Map Capture" and save it as "HELLO" on a freshly initialized disk. When you are finished with a dungeon go to the nearest town to save your status. Return to the dungeon and when you are inside remove the *Phantasie* disk and insert your initialized disk with the new hello. Press **[RESET]**, the disk will reboot and show the picture of the dungeon you have just left. Type in the name of the dungeon, and you have just saved to the disk the picture of the dungeon. Then use your favorite screen dump program to print it.

```

100 REM PHANTASIE MAP CAPTURE
20 POKE - 16304,0: POKE - 16297,0: POKE -
16300,0: POKE - 16301,0
30 VTAB 23: INPUT "NAME OF MAP:" : MS
40 PRINT CHR$(4) "BSAVE" MS : AS2000,LS2000
50 VTAB 24: PRINT : PRINT "PRESS A KEY TO
REBOOT" : GET QS: PR# 6

```

### Checksums

10	- \$BADD	40	- \$3002
20	- \$3588	50	- \$6FAC
30	- \$FA24		

*Phantasie* maps are all on Page 1 of the hi-res pages (including the countryside), so all that this program has to do is make several POKES to display the map, accept the name of it, and BSAVE it to disk.

Now if you find you have missed an area of a dungeon you can return to it easily!

R.D. West

Softkey for...

## Fay: Word Hunter

Didatech Software

### Requirements

- Fast copier that ignores errors
- Sector editor

I managed to deprotect *Fay: Word Hunter* by accident. Track S02, sector S07 is protected and the assumption is that it is used for a nibble count. BLOADing the file *Fay: Word Hunter* showed meaningless information and I assumed that the file was encrypted.

I decided to boot the disk and use an NMI card to interrupt the program after it was completely loaded. The first place I looked was \$BA69-\$BA95 since this unused area of DOS is sometimes used to place nibble count routines. I discovered a JMP to \$6BF2 at \$BA92. I assumed this to be the entry to the





program after everything was loaded. I made a backup copy using *Locksmith Fast Copy* to ignore the protected sector and using the sector editor on *Copy II Plus* scanned the backup for the sequence 4CF26B. I found it on track \$00, sector \$04. I then changed F2 6B to 69 FF. This allowed me to run the program, enter the monitor and examine the program at will. A little experimentation showed that the nibble count appeared to begin at \$8567.

After many hours of hit and miss experimentation, I discovered that changing the value at \$856E from F0 to D0 seemed to bypass or eliminate the nibble count. I can only guess why. Since the program was encrypted the only way to change this byte was to write a short program to insert D0 at \$856E after the program was loaded. I did this just before the JMP to \$6BF2 on track \$00, sector \$04. This did not seem to affect the operation of the program.

**1** Make a backup copy using *Locksmith Fast Copy* or any other method that will ignore the protected sector.

**2** Using a sector editor, make the following edit:

Trk	Sct	Byte(s)	From	To
\$00	\$04	\$89-91	?	A9 D0 8D 6E 85 EA EA EA EA

**Michael A. Horton**

## Alternate Reality Character Editor

This is for all computists who like *Alternate Reality*.

It was difficult to get the information I needed to make the character editor. In an adventure game, the character's statistics are usually kept with the name of the character. I located the names of the characters in track \$00, sector \$01. To my surprise, the "stats" were nowhere to be seen.

I pondered the ways I could try to find them. If part of my character changes then the information that is saved should be different. I wrote a disk comparison program to compare two save disks. I backed up my character disk and played it until something changed. I then saved my character and compared it to the backup copy using the comparison program. The different bytes were what changed.

I did this over and over until I gathered all the "stats" that I wanted. Then I wrote a program to make the whole thing easy. I even put in a help mode, if you need instructions just type ? or /.

Enter the BASIC program and save it as "AR CHAR EDITOR". Then go into the monitor and enter the hex code. Save it as "READ TRACK".

### AR CHAR EDITOR

```

10 REM ALTERNATE REALITY
20 REM CHARACTER EDITOR
30 REM BY MICHAEL HORTON
40 REM
50 PRINT CHR$(13) + CHR$(4) "BLOAD"READ"TRACK"
60 CALL 777: REM CONVERT DOS TO AR DOS
70 CLEAR: RESTORE
80 IO = PEEK(769) + PEEK(770) * 256
90 POKE IO + 1, PEEK(47081): REM GET RIGHT SLOT #
100 CS = "AR"CHAR"DISK": READ NA
110 T = IO + 4: S = IO + 5: CD = IO + 12: PA = IO + 9: BF = 768
120 DIM N(4), NS(4), AS(NA), A(NA), P(NA), AN(NA)
130 FOR I = 1 TO NA
140 READ AS(I), A(I), P(I)
150 NEXT I
160 TEXT: HOME: VTAB 12
170 PRINT "INSERT"ALTERNATE"REALITY"CHARACTER "DISK":
180 POKE - 16368, 0: GET AS
190 IF AS = "2" THEN POKE IO + 2, 2
200 POKE T, 0: POKE S, 1: POKE CD, 1: POKE PA, 32: CALL 774
210 IF PEEK(IO + 13) < > 0 THEN 1800
220 FOR I = 1 TO 12
230 A = PEEK(8191 + I) - 128
240 IF A < > ASC(MID$(CS, I, 1)) THEN PRINT: VTAB 14: PRINT "WRONG"DISK!: FOR ZZ = 1 TO 1500: NEXT ZZ: GOTO 160
250 NEXT I
260 FOR I = 1 TO 4
270 N(I) = PEEK(8207 + I): NS(I) = ""
280 FOR J = 1 TO 32
290 A = PEEK(8319 + J + (I - 1) * 32)
300 IF A < 32 THEN NS(I) = "" : GOTO 330
310 NS(I) = NS(I) + CHR$(A)
320 NEXT J
330 NEXT I
340 TEXT: HOME: VTAB 8: N = 0
350 PRINT "CHOOSE"CHARACTER"TO"EDIT.
360 FOR I = 1 TO 4
370 INVERSE
380 IF N(I) = 16 OR NS(I) = "" THEN NORMAL
390 PRINT I: ".": NS(I)
400 NEXT I: NORMAL
410 PRINT "5...NEW"AR"CHAR"DISK"
420 PRINT "6...EXIT"PROGRAM"
430 PRINT: PRINT "CHOICE:"
440 GET CHS: A = ASC(CH$)
450 IF A = 53 THEN 160
460 IF A = 54 THEN HOME: CALL 780: END
470 IF A = 63 OR A = 47 THEN GOSUB 1490
480 IF N = 0 THEN 520
490 IF A = 4 THEN GOSUB 560
500 IF A = 14 AND NS(N) < > "" THEN GOSUB 610: GOTO 260
510 IF A = 13 AND NS(N) < > "" THEN GOSUB 720:

```

```

GOTO 200
520 IF A < 49 OR A > 52 THEN 340
530 IF N < > 0 THEN 340
540 N = A - 48: PRINT CH$: : GOTO 440
550 REM DELETE/UNDELETE CHARACTER
560 IF N(N) = 16 THEN POKE 8207 + N, 0
570 IF N(N) = 0 THEN POKE 8207 + N, 16
580 POKE T, 0: POKE S, 1: POKE BF, 32: POKE CD, 2: CALL 774
590 N(N) = PEEK(8207 + N): RETURN
600 REM CHANGE CHAR NAME
610 VTAB 19: INPUT "NEW"NAME:" : NNS
620 IF LEN(NNS) > 32 THEN RETURN
630 IF NNS = "" THEN RETURN
640 L = LEN(NNS): FOR I = L TO 32
650 NNS = NNS + " ": NEXT I
660 FOR I = 1 TO 32
670 POKE 8319 + I + (N - 1) * 32, ASC(MID$(NNS, I, 1))
680 NEXT I
690 POKE T, 0: POKE S, 1: POKE CD, 2: POKE BF, 32: CALL 774
700 RETURN
710 REM EDIT CHAR INFO
720 POKE T, N * 4: POKE CD, 1: CALL 771
730 IF PEEK(IO + 13) < > 0 THEN GOTO 1800
740 FOR I = 1 TO NA: AN(I) = 0
750 FOR J = A(I) TO 0 STEP - 1
760 AN(I) = AN(I) + PEEK(P(I) - J) * 256 # J
770 NEXT J
780 NEXT I
790 HOME: PRINT "NAME:" : NS(N)
800 FOR I = 1 TO NA
810 PRINT AS(I): TAB(20)AN(I)
820 NEXT I: POKE 34, 22
830 LN = 1: GOSUB 980
840 A = PEEK(- 16384)
850 IF A < 129 THEN 840
860 POKE - 16368, 0: XN = LN
870 IF A = 136 OR A = 139 THEN LN = LN - 1
880 IF A = 149 OR A = 138 THEN LN = LN + 1
890 IF LN < 1 THEN LN = NA
900 IF NA < LN THEN LN = 1
910 GOSUB 970
920 IF A = 141 THEN GOSUB 1010
930 IF A = 145 THEN GOSUB 1080: RETURN
940 IF A = 191 OR A = 175 THEN GOSUB 1640
950 GOTO 840
960 REM MOVE EDIT LINE
970 VTAB XN + 1: PRINT AS(XN): TAB(20)AN(XN): "*****"
980 INVERSE: VTAB LN + 1: PRINT AS(LN): TAB(20)AN(LN)
990 NORMAL: RETURN
1000 REM CHANGE STAT VALUE
1010 VTAB 23: PRINT "ENTER"NEW"VALUE"(0-" : 256 * 256 # A(LN) - 1: ")":
1020 INPUT "": NV$
1030 IF NV$ = "" THEN HOME: RETURN
1040 NV = VAL(NV$)
1050 IF NV < 0 OR NV < > INT(NV) OR NV > 256 * 256 # A(LN) - 1 THEN HOME: GOTO 1010

```



# Readers Data EXchange



```

1060 AN(LN) = NV: XN = LN: HOME : GOTO 970
1070 REM SAVE STATS
1080 HOME : PRINT "SAVE ABOVE STATISTICS"
      (Y/N) :
1090 GET AS
1100 IF AS = "N" THEN RETURN
1110 IF AS < > "Y" THEN 1090
1120 FOR I = 1 TO NA - 7
1130 FOR J = A(I) TO 0 STEP - 1
1140 P = INT (AN(I) / 256 # J)
1150 AN(I) = AN(I) - P * 256 # J
1160 POKE P(I) - J, P
1170 NEXT J
1180 NEXT I
1190 FOR I = NA - 6 TO NA
1200 FOR J = 2 TO 0 STEP - 1
1210 POKE P(I) - J, AN(I)
1220 NEXT J
1230 NEXT I
1240 POKE CD, 2: CALL 771
1250 RETURN
1260 DATA 21
1270 DATA "CURRENT HIT POINTS" , 2, 11961
1280 DATA "MAXIMUM HIT POINTS" , 2, 11965
1290 DATA EXPERIENCE, 3, 11945
1300 DATA LEVEL, 0, 11949
1310 DATA GOLD, 2, 11969
1320 DATA SILVER, 2, 11973
1330 DATA COPPER, 2, 11977
1340 DATA GEMS, 2, 11981
1350 DATA JEWELRY, 2, 11985
1360 DATA POTIONS, 0, 12157
1370 DATA FOOD, 0, 10667
1380 DATA WATER, 0, 10668
1390 DATA TREASURE FINDING, 0, 10662
1400 DATA UN-NOTICABILITY, 0, 12053
1410 DATA STRENGTH, 0, 12080
1420 DATA INTELLEGEENCE, 0, 12089
1430 DATA WISDOM, 0, 12098
1440 DATA SKILL, 0, 12107
1450 DATA STAMINA, 0, 12116
1460 DATA CHARM, 0, 12125
1470 DATA PHYSICAL SPEED, 0, 12134
1480 REM NAME LIST HELP
1490 HOME : VTAB 3
1500 PRINT "TYPE THE NUMBER OF YOUR CHOICE AND
      THEN"
1510 PRINT "PRESS RETURN, IF YOUR CHOICE WAS
      THE"
1520 PRINT "NAME OF A CHARACTER THEN THAT"
1530 PRINT "CHARACTER'S STATS WILL BE CALLED
      UP AND"
1540 PRINT "YOU WILL BE PUT INTO EDIT MODE."
1550 PRINT : PRINT " IF YOU WISH TO RENAME YOUR
      CHARACTER."
1560 PRINT "TYPE THE APPROPRIATE NUMBER AND
      THEN"
1570 PRINT "TYPE CONTROL N."
1580 PRINT : PRINT " IF YOU WISH TO UNDELETE OR
      DELETE A"
1590 PRINT "CHARACTER TYPE THE APPROPRIATE
      NUMBER"

```

```

1600 PRINT "AND THEN TYPE CONTROL D."
1610 GOSUB 1740
1620 RETURN
1630 REM EDIT STATS HELP
1640 TEXT : HOME : VTAB 3
1650 PRINT " IN ORDER TO MOVE THE INVERSE BAR
      USE THE "
1660 PRINT "ARROW KEYS." : PRINT
1670 PRINT "PRESS RETURN TO EDIT A STAT." :
      PRINT
1680 PRINT " IF YOU DO NOT WISH TO CHANGE A
      STAT'S"
1690 PRINT "VALUE JUST PRESS RETURN." : PRINT
1700 PRINT "TYPE CONTROL Q TO QUIT EDITTING."
1710 GOSUB 1740: POP
1720 GOTO 790
1730 REM WAIT FOR RETURN
1740 PRINT : PRINT "PRESS" : INVERSE
      : PRINT "RETURN" :
1750 NORMAL : PRINT " TO CONTINUE."
1760 GET AS
1770 IF AS < > CHR$ (13) THEN 1760
1780 RETURN
1790 REM ERROR HAS OCCURRED
1800 HOME : VTAB 11
1810 PRINT "AN ERROR HAS OCCURRED WHILE TRYING
      TO"
1820 PRINT "ACCESS THE DISK DRIVE. PLEASE
      RESTART."
1830 END

```

```

300 - $2451
310 - $AA8B
320 - $5052
330 - $6816
340 - $3821
350 - $0DA0
360 - $5C8A
370 - $16AB
380 - $481F
390 - $E5C6
400 - $E086
410 - $0A1E
420 - $7797
430 - $5856
440 - $623C
450 - $7C59
460 - $8AE1
470 - $57EB
480 - $1011
490 - $465D
500 - $FA6D
510 - $EFD3
520 - $7401
530 - $9325
540 - $7DDE
550 - $61B1
560 - $63E3
570 - $875E
580 - $B35A
590 - $5D07
600 - $2460
610 - $0C63
620 - $E3A3
630 - $94F0
640 - $47DE
650 - $66FA
660 - $2C96
670 - $1F39
680 - $1B26
690 - $A20A
700 - $7642
710 - $551E
720 - $17F3
730 - $6CB1
740 - $0643
750 - $48C3
760 - $F332
770 - $88CA
780 - $0733
790 - $AEB7
800 - $A231
810 - $A008
820 - $9D21
830 - $A887
840 - $268F
850 - $3E9C
860 - $62AA
870 - $9CE1
880 - $DEF9
890 - $0542
900 - $D224
910 - $62E2
920 - $B4C6

```

```

1220 - $EB2B
1230 - $384B
1240 - $47F2
1250 - $3156
1260 - $6BE6
1270 - $B83A
1280 - $5DC9
1290 - $A188
1300 - $0017
1310 - $3ED3
1320 - $2878
1330 - $9131
1340 - $26FE
1350 - $3585
1360 - $654E
1370 - $E30A
1380 - $A880
1390 - $6F98
1400 - $BE27
1410 - $8133
1420 - $37BA
1430 - $DAAE
1440 - $0310
1450 - $1C89
1460 - $672B
1470 - $54DE
1480 - $9C90
1490 - $8537
1500 - $445F
1510 - $82CB
1520 - $C48B
1530 - $6E42
1540 - $1D0A
1550 - $19AA
1560 - $417F
1570 - $819C
1580 - $16B6
1590 - $B60B
1600 - $83A9
1610 - $5576
1620 - $E831
1630 - $7066
1640 - $295E
1650 - $3229
1660 - $7D9E
1670 - $2F61
1680 - $D877
1690 - $5719
1700 - $37DE
1710 - $AE10
1720 - $CE32
1730 - $3FA8
1740 - $0B99
1750 - $95C6
1760 - $95F9
1770 - $8487
1780 - $D556
1790 - $E371
1800 - $E9ED
1810 - $3A55
1820 - $A9C2
1830 - $9791

```

## Checksums

```

10 - $BADD 930 - $E36F
20 - $9B13 940 - $2EBF
30 - $4D3B 950 - $3709
40 - $AD92 960 - $20C3
50 - $352C 970 - $224A
60 - $77A4 980 - $68C4
70 - $6EAD 990 - $CA50
80 - $55BB 1000 - $553C
90 - $D68B 1010 - $0AD4
100 - $2A07 1020 - $78A1
110 - $2D75 1030 - $BF0E
120 - $8045 1040 - $89C6
130 - $3FAF 1050 - $E6D1
140 - $25CD 1060 - $4AE7
150 - $522F 1070 - $2B3A
160 - $1D94 1080 - $F999
170 - $D349 1090 - $8910
180 - $B31F 1100 - $6D36
190 - $81B9 1110 - $888B
200 - $878D 1120 - $3029
210 - $1242 1130 - $EEEE
220 - $E044 1140 - $6813
230 - $A130 1150 - $D420
240 - $934A 1160 - $3A5D
250 - $EAB4 1170 - $781C
260 - $DD5E 1180 - $7270
270 - $FD1F 1190 - $418D
280 - $61BD 1200 - $D13D
290 - $BAD2 1210 - $765E

```





## READ TRACK

0300:	20 4D 03 4C 21 03 4C 3E	\$EB2F
0308:	03 4C 62 03 4C 7B 03 4C	\$E409
0310:	94 03 00 00 00 00 00 00	\$9FAA
0318:	00 00 00 00 00 00 00 00	\$AF0A
0320:	00 A9 0F 8D 52 03 18 6D	\$4330
0328:	00 03 8D 56 03 20 3E 03	\$2DCF
0330:	AD 5A 03 D0 08 CE 56 03	\$C060
0338:	CE 52 03 10 F0 60 A9 03	\$691E
0340:	A0 4D 20 D9 03 B0 05 A9	\$994A
0348:	00 8D 5A 03 60 01 60 01	\$966B
0350:	00 00 00 5E 03 00 20 00	\$0555
0358:	00 01 00 00 60 01 00 01	\$3065
0360:	D8 EF EE 29 BA CE 2A BA	\$8DF6
0368:	EE 49 BA CE 4A BA EE 96	\$5C65
0370:	BA CE 97 BA EE D6 BA CE	\$369A
0378:	D7 BA 60 CE 29 BA EE 2A	\$B63A
0380:	BA CE 49 BA EE 4A BA CE	\$53BD
0388:	96 BA EE 97 BA CE D6 BA	\$00A9
0390:	EE D7 BA 60 A9 00 85 00	\$3F0A
0398:	A9 20 85 01 A0 00 B1 00	\$6403
03A0:	49 87 91 00 C8 D0 F7 E6	\$FCDA
03A8:	01 A5 01 C9 30 D0 ED 60	\$E607

Charles Taylor

Softkey for...

## Mastertype v2.1

Scarborough

### Requirements

- Apple IIe or IIc with 128K
- Super IOB 1.5 with New Swap Controller
- Sector editor with search feature (Copy II+)
- DOS 3.3 System Master disk
- One blank disk side
- XFER.BOOT & RESTORE (COMPUTIST #28) or other way into the monitor.

Mastertype 2.1 is a double hi-res update of the classic Mastertype typing teacher game. It is not a difficult program to copy, but the copies have a habit of dying or becoming particular of which disk drives will boot it.

The excellent softkey presented way back in COMPUTIST #15 for Mastertype 1.7 will not work on version 2.1, but the information necessary to crack 2.1 is all there.

I resolved to make an unprotected version of 2.1 that would boot under standard DOS. I missed that goal, but at least I unprotected it.

Mastertype's hello program is a text file instead of an Applesoft program, so we must change DOS to execute (EXEC instead of RUN) the hello file.

**1** Boot a DOS 3.3 system master and fill memory from \$BA69 to \$BA94 with 60s.

CALL-151

BA69:60  
BA6A<BA69.BA94M

**2** Insert a blank disk and initialize it to create a slave disk.

FP  
INIT HELLO

**3a** Now, we need to capture the Master Type RWTS. If you do not have a copy card or other way to force entry into the monitor, then skip to step 3b. Otherwise, boot your original. When the drive stops, enter the monitor and move the RWTS to a safe place in memory, then boot your slave disk and skip to step 4.

1900<B800.BFFFM

**3b** Insert the disk with XFER.BOOT and RESTORE into the drive. Load XFER.BOOT.

BLOOD XFER.BOOT

**3c** Put your original disk into the drive and boot it into upper memory.

PR#3  
CALL 768

**3d** When the drive stops, press **RESET**. Insert the disk with XFER.BOOT and RESTORE into the drive and use RESTORE to move the trapped RWTS into main memory.

CALL-151  
3F8: 4C 00 03  
BLOOD RESTORE  
1900<B800.BFFF  Y

**4** Save the protected RWTS to your Super IOB disk.

BSAVE RWTS.MASTERTYPE2.1, A\$1900,  
L\$800

**5** Load Super IOB 1.5 (from your Starter Kit) and merge the standard swap controller with it. Change line 10010 to load the RWTS.

LOAD SUPER IOB  
EXEC SWAP.CON  
10010 PRINT CHR\$(4) "BLOOD  
RWTS.MASTERTYPE2.1"

RUN

**6** Using a sector editor with a search feature, search for all occurrences of the text string "YZ123" and change to "BLOOD", and the text string "YZ23" and change to "BRUN". The tricky part here is that the "search text" will only find some of the YZ stuff. Some of the YZ's and all the KILLDES and CATNDOGS are in flashing text. For these you will have to search for the equivalent hex bytes. Similarly, change "CATNDOG" to "CATALOG", and "KILLDE" to "DELETE". I did not find any SAVES that

needed to be changed to INIT. The table below lists all of the commands that need to be corrected.

## BLOOD and BRUN

Trk	Sct	Byte(s)	From	To
\$10	\$04	\$46	YZ123	BLOOD
		\$7D	YZ123	BLOOD
		\$88	YZ23	BRUN
\$1F	\$0D	\$04	YZ123	BLOOD
		\$1E	YZ123	BLOOD
		\$3C	YZ123	BLOOD
		\$58	YZ123	BLOOD
		\$72	YZ123	BLOOD
		\$83	YZ123	BLOOD
		\$96	YZ123	BLOOD
		\$AB	YZ123	BLOOD
		\$C1	YZ123	BLOOD
		\$D6	YZ123	BLOOD
\$0E	\$EA	YZ123	BLOOD	BLOOD

## Flashing BRUN

Trk	Sct	Byte(s)	From	To
\$0A	\$04	\$C7	59 5A 32 33	42 52 55 46
\$0E	\$0F	\$40	59 5A 32 33	42 52 55 46
\$15	\$09	\$AC	59 5A 32 33	42 52 55 46

## Flashing BLOOD

Trk	Sct	Byte(s)	From	To
\$20	\$0E	\$EA	59 5A 31 32 33	42 4C 4F 41 44

## Flashing CATALOG

Trk	Sct	Byte(s)	From	To
\$15	\$0D	\$B3	4E 44	41 4C
\$15	\$0B	\$AC	4E 44	41 4C
\$0A	\$02	\$86	4E 44	41 4C

## Flashing DELETE

Trk	Sct	Byte(s)	From	To
\$15	\$00	\$13	4B 49 4C 4C 44	44 45 4C 45 54
			45	45
\$15	\$0A	\$74	4B 49 4C 4C 44	44 45 4C 45 54
			45	45

Jerry Stevens

Softkey for...

## Create with Garfield

DLM

I was asked to make a backup of Create with Garfield for our school librarian. (Can you imagine giving primary school age children an original disk to use?) In COMPUTIST #52, there is a reference to an earlier article by Mr. Brown. I agree that the response in COMPUTIST #44 is nonsense. (I wonder if it



was an April Fools letter?) Anyway, there was no help there. The *Create with Garfield* that I was working with is called the *Deluxe Version* and most likely is different than those referred to by others, but here is a crack that seems to work on this particular disk. Maybe the protection is similar on the other versions.

The protection apparently is a signature on track \$22. This track is the only one not in standard format. I found the code that checks for the signature is in memory from \$6D3E to \$6E64. It is part of a file called "LØ". They use the direct SCØEC rather than the usual indirect SCØ8C,X to read a byte from the disk. Don't bother looking on the disk for it however, as it is stored there as \$1FEC instead. The code around \$6D4A then changes the 1F to a CØ. It appeared that all failures were sent to \$6DE2 where the disk drive is turned off, the carry set, and then a return. If the signature is found, and I don't pretend to understand all they go through there, then the carry is cleared.

If you look at the bytes following \$6E64 you will find some of the data it uses. (Lots of \$F7, \$F6, \$EF etc. patterns, and if you nibble read track \$22 you will find the same bytes.) I wasn't sure just where to try to mess up the check since it enters the routine at several different places. A return at the beginning would just be bypassed. How about just clearing the carry instead of setting it on failure in byte \$6DE5? It works! Who says you have to know what you are doing, you just need to be lucky!

To back up the disk use any copier that allows you to specify tracks and copy tracks \$ØØ - \$21. I like to format the disk first so that track \$22 will be formatted even if it isn't used. That way any copier will work without hanging up on track \$22. By the way, track \$23 seems to have a signature of some sort on it also, but apparently isn't used on this disk.

After copying, then search the disk with a sector editor for ADE8 1F 38 6Ø A2 ØØ ADE C 1F and change the 38 to a 18. I found it on track \$15, sector \$ØF, byte \$E9. An alternate method is to boot a DOS 3.3 disk (to get control), then re-insert your copy and:

**BLOAD LØ,A\$6ØØØ**

**CALL-151**

**6DE5:18**

*(was \$38)*

**BSAVE LØ,A\$6ØØØ,L\$134Ø**

By the way, you can copy all the files to any DOS 3.3 disk, except they will not all fit. They have modified the VTOC to crowd things onto the disk. If you want to create some additional room on your disk then you will need to modify your disk to open up some unused sectors (see COMPUTIST #3Ø). I see no advantage to this as they have already used a fast DOS on the original. Too bad it isn't ProDOS, then it could go on a 3.5" disk. I'm not sure the *Create with Garfield* mentioned in COMPUTIST #52 is the same as mine but hopefully this will be of some help. Maybe

someone with better understanding can take this and improve upon the crack. In the meantime have fun creating your cartoons.

P.S. I have to agree with Jim Hart about Utilico and their E.D.D. I have version 4 and, in addition to their protection, I object to not being given the parm locations and uses so I can customize it. For an additional \$34 I can purchase the source code and description of the parms used. I paid nearly \$1ØØ for a program and then find they will sell me at additional cost the information I need to make full use of it? Fooey, I recommend *Copy II Plus!* It is not protected, is one third the price, and they tell you what the various parameters do without any extra charge. (And it works as good or maybe even better!)

P.P.S Sorry to see you adding different computers. I expect that is a sign of a dying magazine. Too bad, I think you had a great thing going. I do not own an IBM, nor do I plan on doing so. Why should I have to clutter up my shelves with worthless information? It is upsetting to find a program that you have been looking for only to discover it is for the IBM version not the Apple. If there are rich people out there who can afford both computers then I suspect they could also afford two magazines! Keep it pure!

---



---

Rene Gaudet

---



---

## ProDOS Block Editor Adding a Search Command

### ■ Requirements

- 64K Apple with ProDOS
- ProDOS Block Editor (COMPUTIST #26)

I am familiar with DOS 3.3 because that is what I used on my Apple II Plus. I never felt the need to convert to ProDOS until I purchased an Apple IIgs. All of a sudden, sector edits on a ProDOS diskette with a DOS 3.3 editor became a nightmare of mapping logical sectors to blocks, not to mention that I could not do a thing with my 3.5" drive.

I went through some back issues of COMPUTIST to see what other people used to edit ProDOS diskettes. In COMPUTIST #26, there was an article regarding a program written by Bob Bergstrom. The program is called *ProDOS Block Editor*, and it is a "zap" utility for the ProDOS operating system. The program is modeled after the *Diskedit* program for the DOS 3.3 operating system. *ProDOS Block Editor* was just what I was looking for.

After using it a while, I found a couple of things I didn't like: It would not work past ProDOS block 279 and it did not have a search capability.

Fixing the first problem was a breeze. I changed a couple lines in the BASIC portion of the program. I added a new structure MB which is a two dimensional array that contains the maximum number of blocks for a given slot and drive.

Implementing a disk search was more complicated, but Bob Bergstrom had done all the hard work.

### Applesoft -vs- Assembler

I considered doing the whole thing in BASIC, but discarded that notion because I hate to wait. I decided to do the front end and back end of the find command in BASIC and the guts of the search in assembly. I wrote and tested the assembly portion in a afternoon, but interfacing it to the BASIC portion took two days.

First, I had to learn the BASIC portion of the program and figure a way to stick my wart (Find Command) on it. Personally, I hate BASIC, the variable names are cryptic, and any program over a page long is impossible to follow (No offense Bob B., it's the language, not the programmer). I wish COMPUTIST would take the same approach as Nibble when publishing programs, I like to see some verbage on the general program structure and uses of variables.

### Implementing a Find Command

Whenever possible, I used existing code from the original program. I placed the BASIC portion at the end of the program and the assembly portion on page \$8F. There was not enough room on page \$Ø3 and the program was already modifying HIMEM.

When the user selects the "F" option, it invokes the find byte pattern command. The user is prompted for the starting block number, and then the ending block number of the search. The user is then prompted to enter a byte pattern. The maximum number of bytes that can be entered is 16. When prompted to enter the byte pattern for the search, the user is really in edit mode, so whatever display criteria was selected prior to executing "F" find is what kind of input is expected (If hex mode is selected, Find accepts hex digits. If ASCII mode is selected, Find accepts ASCII characters). The ESC key is used to exit input mode of the byte pattern. The values supplied by the user are verified and POKEd into locations for the assembly portion. The assembly portion is CALLED, and when the assembly portion finishes, the BASIC portion checks a result flag and acts accordingly. If the byte pattern was not found, an error message is displayed. If the byte pattern was found, the block is displayed





to the user with the cursor located on the first byte of the requested byte pattern.

One exception to this is when a byte pattern spans two ProDOS blocks, the cursor is located on the last byte of the byte pattern. Finally, the user is returned to command mode and all commands are available to the user.

## Bugs

There are two bugs that I know about with the code I added. I will leave it to some whiz-bang BASIC programmer to fix them.

1) There are characters left on the screen when ProDOS block numbers take four or more character positions. This becomes obvious when editing a 3.5" disk and decimal mode is selected. I use only hex mode, so I didn't bother fixing the bug.

2) You can't abort the FIND command once you start it.

## Entering the code

**1** Enter the BASIC program and save it to disk.

### SAVE PROEDIT.FIND

**2** Enter the ProEdit machine code and save it.

### BSAVE OBJ.PROEDIT, A\$300, L\$A5

**3** Enter the machine code search routine and save it.

### BSAVE OBJ.PROED.FIND, A\$8F00, L\$8D

## PROEDIT.FIND

```

10 REM *
20 REM * PROEDIT
30 REM * PRODOS BLOCK
40 REM * EDITOR
50 REM * BY BOB BERSTROM
60 REM * (C) 1985
70 REM * SOFTKEY PUBLISHING
80 REM *
82 REM * FIND BYTE PATTERN ON DISK
85 REM * MODIFICATION
90 REM * BY RENE GAUDET. 1/88
95 REM *
97 REM
100 TEXT : HOME : HIMEM : 35000 : GOSUB 1940 : GOSUB
550 : GOTO 1080
110 REM WAIT FOR KEY
120 KY = PEEK (KB) : IF KY < C7 THEN 120
130 POKE KS,0 : RETURN
140 REM MOVE CURSOR IN FIELD
150 GOSUB 270 : PRINT "" : KY = KY - 8 : ON KY GOTO
160, 180, 200, 1080, 230
160 VT = VT - 1 : IF VT < 0 THEN VT = 19 : GOTO 240
170 GOTO 250
180 HT = HT - 1 : IF HT < 0 THEN HT = 13 : GOTO 160
190 GOTO 250
200 HT = HT + 1 : IF HT > 13 THEN HT = 1 : GOTO 230
210 IF VT = 19 AND HT > 9 THEN HT = 1 : GOTO 230

```

```

220 GOTO 250
230 VT = VT + 1 : IF VT > 19 THEN VT = 0
240 IF VT = 19 AND HT > 9 THEN HT = 9
250 GOSUB 270 : INVERSE : PRINT ">" : NORMAL : PT
= VT * 13 + HT - 1 + BF * C2 : RETURN
260 REM CALC CURSOR POSITION
270 IF HT < 1 THEN HT = 13 : VT = VT - 1 : IF VT <
0 THEN VT = 19 : HT = 9
280 HTAB (HT - 1) * 3 + 1 : VTAB VT + 2 : RETURN
290 REM PRINT CURSOR POSITION
300 KY = ABS (KY) : GOSUB 270 : PRINT "" : VT = INT
(KY / 13) : HT = KY - (VT * 13) + 1 : GOSUB
550 : GOTO 250
310 REM PRINT "___"
320 FOR I = 1 TO 40 : PRINT "-" : NEXT : RETURN
330 REM PROCESS EDIT DATA
340 NORMAL : GOSUB 270 : PRINT ">" CHR$ (8) CHR$
(8) : IF HF = 1 THEN 370 : REM HEX DATA ??
350 GOSUB 460 : IF NOT HB THEN KY = KY - C7 : REM GET
ASCII CHAR, SET BIT 7
360 GOTO 500
370 GOSUB 830 : IF KY > 15 THEN PRINT CHR$ (7) :
GOTO 370 : REM 1ST HEX DIGIT
380 A1 = KY : GOSUB 420
390 GOSUB 830 : IF KY > 15 THEN PRINT CHR$ (7) :
GOTO 390 : REM 2ND HEX DIGIT
400 A2 = KY : GOSUB 420 : GOSUB 900 : GOTO 500 : REM
CONVERT DIGITS TO TOTAL
410 REM PRINT DEC/HEX DIGIT FROM LOOKUP TABLE
420 PRINT MID$ (HES, KY + 1, 1) : RETURN
430 REM CALC A1 MOD A2
440 NU = INT ( ABS (A1) - INT ( ABS (A1 / A2) ) *
ABS (A2) + .05) * SGN (A1 / A2) : RETURN
450 REM INPUT ASCII CHAR
460 GOSUB 120 : IF KY < > 155 THEN RETURN
470 BL = PEEK (BK + 1) * C2 + PEEK (BK) : GOSUB
1920 : GOSUB 550 : GOSUB 600 : GOSUB 530 : POKE
CM, RD : ONERR GOTO 1080
475 IF SE = 1 THEN ONERR GOTO 2100
480 POP : GOTO 480
490 REM POKE VALUE INTO BUFFER. PRINT TO FIELD
500 POKE BS + PT, KY : GOSUB 270 : PRINT "" : POKE
TM, KY : CALL DP : GOTO 200
510 REM ACCESS DISK. PRINT BUFFER FIELD
520 CALL PD
530 CALL PB : GOTO 250
540 REM PRINT STATUS LINE LABELS
550 VTAB 23 : HTAB 1 : CALL - 958 : INVERSE : PRINT
"BLOCK" : IF NOT BF THEN HTAB 12 : NORMAL :
PRINT ">" : INVERSE : PRINT "BUF" :
NORMAL : PRINT "FER" : INVERSE
560 IF BF THEN HTAB 13 : NORMAL : PRINT "BUF" :
INVERSE : PRINT "FER" : NORMAL : PRINT "<"
: INVERSE
570 HTAB 33 : PRINT "PTR" : PRINT "SLOT" : HTAB
9 : PRINT "DRIVE" : HTAB 18 : PRINT "I/O" :
580 VTAB 24 : HTAB 27 : FLASH : PRINT "COMMAND" :
NORMAL : PRINT "" : RETURN
590 REM PRINT STATUS LINE VALUES

```

```

600 VTAB 23 : HTAB 7 : IF CF THEN PRINT BL "" :
HTAB 37 : PRINT PT "" : VTAB 24 : HTAB 36 :
IF PT - OF > - 1 THEN PRINT "" : CHR$
(8) : CHR$ (8) : CHR$ (8) : PT - OF :
GOTO 700
610 IF CF THEN PRINT "" : CHR$ (8) : CHR$ (8) :
CHR$ (8) : CHR$ (8) : PT - OF : GOTO 700
620 REM
625 PRINT INT (BL / C2) : POKE TM, BL - (INT (BL
/ C2) * C2) : CALL HP : GOTO 640
630 REM
640 HTAB 37 : IF PT > C6 THEN PRINT "1" : POKE
TM, PT - C2 : CALL HP : GOTO 660
650 PRINT "0" : POKE TM, PT : CALL HP
660 VTAB 24 : HTAB 36 : PRINT "" : CHR$ (8) : CHR$
(8) : CHR$ (8) : CHR$ (8) : IF PT - OF > - 1
THEN PRINT "" : GOTO 680
670 PRINT "-" :
680 IF ABS (PT - OF) > C6 THEN PRINT "1" : POKE
TM, ABS (PT - OF) - C2 : CALL HP : GOTO 700
690 PRINT "0" : POKE TM, ABS (PT - OF) : CALL HP
700 VTAB 23 : HTAB 21 : PRINT "" : INVERSE : IF
HF = 1 THEN PRINT "HEX" : NORMAL : PRINT "#
*****" :
710 IF HF = 2 AND NOT HB THEN PRINT "ASC/MSB=" :
NORMAL : PRINT "0" :
720 IF HF = 2 AND HB THEN PRINT "ASC/MSB=" :
NORMAL : PRINT "1" :
730 VTAB 24 : HTAB 6 : PRINT SL : HTAB 15 : PRINT
DR : HTAB 22 : IF NOT CF THEN PRINT "HEX" :
RETURN
740 PRINT "DEC" : RETURN
750 REM RING BELL
760 PRINT CHR$ (7) CHR$ (7) : RETURN
770 REM DECODE ERROR. PRINT TO FIELD
780 ER = PEEK (EF) : POKE 34, 1 : POKE 35, 21 : HOME
: VTAB 12 : HTAB 12 : IF ER = 39 THEN PRINT
"DISK I/O ERROR"
790 IF ER = 40 THEN PRINT "NO DEVICE CONNECTED"
800 IF ER = 43 THEN PRINT "DISK WRITE PROTECTED"
810 FOR I = 1 TO 2500 : NEXT : POKE 772, 0 : POKE
35, 24 : GOTO 530
820 REM KEYPRESS DECODE FOR NUMERICS
830 GOSUB 460 : IF KY = 141 THEN RETURN
840 KY = KY - 176 : IF KY < 0 OR KY > 22 THEN GOSUB
760 : GOTO 830
850 IF KY > 9 THEN KY = KY - 7 : IF KY < 10 OR KY
> 15 THEN GOSUB 760 : GOTO 830
860 IF ED THEN RETURN : REM EDIT MODE? THEN MUST
BE HEX
870 IF CF AND KY > 9 THEN GOSUB 760 : GOTO 830 :
REM COMMAND LINE DEC NUMBER TOO BIG
880 RETURN
890 REM ASSEMBLE NUMBER
900 IF ED OR NOT CF THEN KY = A1 * C3 + A2 : RETURN
: REM 'EDIT' OR COMMAND LINE HEX (2 DIGITS)
910 IF NOT ED AND CF THEN KY = A1 * C9 + A2 : RETURN
: REM NOT 'EDIT'. GET DEC NUMBER (2 DIGITS)
920 IF ED OR NOT CF THEN KY = A1 * C2 + A2 * C3
+ A3 : RETURN : REM 'EDIT' OR COMMAND LINE
HEX (3 DIGITS)

```



# Readers Data EXchange



```

930 KY = A1 * C8 + A2 * C9 + A3: RETURN : REM
    COMMAND LINE DEC (3 DIGITS)
940 REM ENCODE STRING INPUT TO HEX VALUE
950 AB$ = "000" + AB$: AB$ = RIGHTS (AB$, 4): J =
    1: FOR I = 1 TO 4: PR(I) = 0: NEXT I: FOR I =
    1 TO LEN (AB$): PR(I) = ASC ( MID$
    (AB$, J, 1)) - 48: IF PR(I) > 9 THEN PR(I) =
    PR(I) - 7
960 J = J + 1: NEXT : PR = C1 * PR(1) + C2 * PR(2)
    + C3 * PR(3) + PR(4): RETURN
970 REM PRODOS BLOCK NUMBER
980 VTAB 23: HTAB 1: FLASH : PRINT "BLOCK" ::
    NORMAL : PRINT "" :: GOSUB 830: IF KY > 15
    THEN KY = PEEK (BK) + PEEK (BK + 1) * C2:
    GOTO 1020
990 A1 = KY: GOSUB 420: GOSUB 830: IF KY > 15 THEN
    KY = A1: GOTO 1020
1000 A2 = KY: GOSUB 420: GOSUB 830: IF KY > 15
    THEN GOSUB 900: GOTO 1020
1010 A3 = KY: GOSUB 420: GOSUB 920
1020 IF (KY < 0) OR (KY > MB(SL, DR)) THEN PRINT
    CHRS (7) :: GOTO 980
1030 BL = KY: GOSUB 600
1040 VTAB 23: HTAB 1: INVERSE : PRINT "BLOCK" ::
    NORMAL
1045 IF SE = 1 THEN RETURN
1050 IF PEEK (CM) = 129 THEN VTAB 24: HTAB 27:
    FLASH: PRINT "??WRITE??" CHRS (7) CHRS (7)
    CHRS (7) :: NORMAL : GOSUB 460: IF KY < >
    141 THEN 470
1060 POKE BK + 1, INT (BL / C2): POKE BK, INT (BL
    - (INT (BL / C2) * C2)): GOTO 520
1070 REM TOP OF MAIN "COMMAND" LOOP
1080 POKE 216, 0: ED = 0: GOSUB 580: GOSUB 600: IF
    PEEK (772) > 0 THEN GOSUB 780
1090 REM GET COMMAND
1100 GOSUB 270: GOSUB 120: IF KY = 140 THEN GOSUB
    1330: GOTO 1080
1110 IF KY = 142 THEN GOSUB 1360: GOTO 1080
1120 KY = KY - 192: IF KY = -1 THEN 1310
1130 IF KY > 0 AND KY < 27 THEN ON KY GOSUB 1160
    , 1550, 1200, 1230, 1260, 2000, 760, 1570
    , 150, 150, 150, 1710, 150, 760, 1630, 1390
    , 760, 1470, 1490, 760, 1290, 760, 1530, 1610
    , 760, 1590
1140 GOTO 1080
1150 REM TOGGLE ASCII / HEX FIELD DISPLAY
1160 HX = NOT HX: IF HX THEN 1180
1170 HF = 2: POKE FM, HF: GOTO 530
1180 HF = 1: POKE FM, HF: GOTO 530
1190 REM TOGGLE CTL-CHAR PRINT
1200 IF HF = 2 THEN CC = NOT CC: POKE CT, CC: GOTO
    530
1210 RETURN
1220 REM TOGGLE ACTIVE DRIVER
1230 DR = (NOT (DR - 1)) + 1
1240 POKE UT, (DR - 1) * C7 + SL * C3: GOTO 550
1250 REM EDIT
1260 VTAB 24: HTAB 27: FLASH : PRINT CHRS (7)
    ">>EDIT<<" :: ED = 1
1270 GOSUB 340: GOSUB 600: GOTO 1270
1280 REM TOGGLE COM'D LINE HEX/DEC
1290 CF = NOT CF: RETURN
1300 REM HELP SCREEN]
1310 GOSUB 1860: GOSUB 550: GOSUB 530: GOTO 1080
1320 REM LAST BLOCK
1330 BL = BL - 1: IF BL > = 0 THEN 1060
1340 BL = MB(SL, DR): GOTO 1060
1350 REM NEXT BLOCK
1360 BL = BL + 1: IF BL < = MB(SL, DR) THEN 1060
1370 BL = 0: GOTO 1060
1380 REM SET POINTER IN BUFFER
1390 VTAB 24: HTAB 27: INVERSE : PRINT "SET-PTR"
    :: VTAB 23: HTAB 33: FLASH : PRINT "PTR"
    CHRS (7) :: NORMAL : PRINT "" CHRS (8)
    CHRS (8) CHRS (8) :: GOSUB 830: IF KY > 15
    THEN GOSUB 550: GOTO 250
1400 A1 = KY: GOSUB 420: GOSUB 830: IF KY > 15
    THEN KY = A1: GOTO 300
1410 A2 = KY: GOSUB 420: GOSUB 830: IF KY > 15
    THEN GOSUB 900: GOTO 300
1420 A3 = KY: GOSUB 420: GOSUB 920: IF NOT BF THEN
    IF KY > C6 THEN GOSUB 760: GOTO 1390
1430 IF BF THEN IF KY < C2 OR KY > C5 THEN GOSUB
    760: GOTO 1390
1440 IF KY > C6 THEN KY = KY - C2
1450 GOTO 300
1460 REM READ A BLOCK FROM DISK
1470 VTAB 24: HTAB 27: INVERSE : PRINT ">READ<"
    :: NORMAL : PRINT CHRS (7) "" :: GOSUB 980:
    GOTO 550
1480 REM SET ACTIVE SLOT
1490 VTAB 24: HTAB 27: INVERSE : PRINT "SET"
    "SLOT" :: HTAB 1: FLASH: PRINT "SLOT" CHRS
    (7) :: NORMAL : HTAB 6: PRINT "" :: GOSUB
    830: IF KY > 15 THEN GOTO 550
1500 IF KY < 1 OR KY > 7 THEN 1490
1510 SL = KY: POKE UT, (DR - 1) * C7 + SL * C3:
    GOTO 550
1520 REM WRITE BLOCK TO DISK
1530 POKE CM, WR: VTAB 24: HTAB 27: INVERSE :
    PRINT ">WRITE<" CHRS (7) :: NORMAL :
    GOSUB 980: POKE CM, RD: GOTO 550
1540 REM TOGGLE DISPLAYED BUFFER HALF
1550 BF = NOT BF: PT = VT * 13 + HT - 1 + BF * C2:
    POKE BU, 144 + BF: GOSUB 530: GOTO 550
1560 REM TOGGLE ASCII HI-BIT (EDIT MODE)
1570 HB = NOT HB: RETURN
1580 REM ZERO POINTER OFFSET
1590 OF = PT: RETURN
1600 REM EXIT TO BASIC
1610 TEXT : HOME : END
1620 REM OUTPUT TO PRINTER
1630 VTAB 23: HTAB 1: PRINT "" VERIFY PRINTER"
    ON "&PRESS" <RETURN> ""
1640 GOSUB 120
1650 VTAB 1: GOSUB 1840: GOSUB 600: GOSUB 250:
    VTAB 1: PRINT : IF KY < > 141 THEN RETURN
1660 REM SET UP FOR PARALLEL PRINTER CARD
1670 PRINT DS: "PR#1": PRINT IS: "80N"
1680 FOR V = 1 TO 24: VTAB V: X = PEEK (40) + PEEK
    (41) * 256: FOR H = X TO X + 39: AS = PEEK
    (H): PRINT CHRS (((AS < 32) * (AS + 64))
    + ((AS > 31) * AS)) :: NEXT : PRINT : NEXT
1690 FOR I = 1 TO 5: PRINT : NEXT : PRINT DS "PR#0"
    : RETURN
1700 REM DISASSEMBLE BUFFER
1710 TEXT : HOME : INPUT "STARTING BYTE (^$0^TO^
    SIFE^HEX):" : ABS: IF LEN (ABS) = 0 THEN PR
    = 0: GOTO 1740
1720 GOSUB 950: IF PR < 0 OR PR > 510 THEN 1710
1730 REM PRINT LOOP STARTS HERE
1740 A1 = BS + PR: A2 = C2: GOSUB 440: POKE PC, NU:
    POKE PC + 1, INT ((BS + PR) / C2): CALL ID
1750 A1 = PEEK (BS + PR): A2 = 32: GOSUB 440: IF
    NU < > C3 THEN 1790
1760 HTAB 27: PRINT "(" : NU = PEEK (BS + PR + 1):
    IF NU < C7 THEN 1780
1770 PRINT NU - 254; ")" : GOTO 1790
1780 PRINT "+" : NU + 2; ")" :
1790 PR = PEEK (LE) + PR + 1: IF PR > C5 THEN PRINT
    : PRINT DS: "PR#0": PRINT : INPUT ""
    PLEASE PRESS <RETURN> " : ABS: GOSUB
    1920: GOSUB 550: CALL PB: GOTO 250
1800 IF PEEK (KB) < C7 THEN 1740
1810 REM PRINT LOOP END - IF YOU'RE HERE KEY WAS
    PRESSED
1820 KY = PEEK (KB): POKE KS, 0: IF KY < > 160 AND
    KY < > 141 THEN 1740: REM SPACE OR
    RETURN?
1830 GOSUB 120: IF KY < > 141 THEN 1740
1840 GOSUB 1920: GOSUB 550: CALL PB: GOTO 250
1850 REM PRINT HELP TABLE
1860 PRINT : TEXT : HOME : HTAB 8: PRINT "COMMAND"
    TABLE:" : PRINT : HTAB 4: PRINT "A^-^TOGGLE^
    ASCII/HEX^DISPLAY" : HTAB 4: PRINT "B^-^
    TOGGLE^ACTIVE^BUFFER^HALF"
1870 HTAB 4: PRINT "C^-^TOGGLE^CONTROL^CHAR"
    PRINT" : HTAB 4: PRINT "D^-^SET^DRIVE" :
    HTAB 4: PRINT "E^-^EDIT^(ESC^TO^END)"
1875 HTAB 4: PRINT "F^-^FIND^BYTE^PATTERN"
1880 HTAB 4: PRINT "H^-^TOGGLE^ASCII^EDIT"
    HI-BIT" : HTAB 4: PRINT "I, J, K, M^-^MOVE"
    CURSOR^ IN^ FIELD" : HTAB 4: PRINT "L^-^
    DISASSEMBLE^BUFFER" : HTAB 4: PRINT "O^-^
    OUTPUT^SCREEN^TO^PRINTER"
1890 HTAB 4: PRINT "P^-^POSITION^CURSOR^ IN^
    FIELD" : HTAB 4: PRINT "R^-^SET^BLOCK^THEN"
    READ"" (RETURN^ USES^
    DISPLAYED^VALUES)" : HTAB 4: PRINT "S^-^
    SET^SLOT" : HTAB 4: PRINT "U^-^UPDATE"
    COMMAND^LINE^DEC/HEX"
1900 HTAB 4: PRINT "W^-^WRITE^BLOCK^TO^DISK" :
    HTAB 4: PRINT "X^-^EXIT^TO^APPLESOFT" :
    HTAB 4: PRINT "Z^-^ZERO^RELATIVE^BYTE"
    COUNTER" : PRINT "CTRL^L/^N^-^LAST/^NEXT"
    SECTOR"
1910 VTAB 24: HTAB 8: PRINT "PRESS^ANY^KEY^TO"
    CONTINUE" :: GOSUB 460
1920 POKE 34, 0: HOME : GOSUB 320: VTAB 22: GOSUB
    320: RETURN
1930 REM POKE ML 1/0, INIT VARIABLES, SET
    CONSTANTS
1940 IF PEEK (774) < > 32 THEN PRINT CHRS (4)
    "BLOAD^OBJ.^PROEDIT, A$300"
1945 PRINT CHRS (4) "BLOAD^OBJ.^PROED.FIND,
    A$8F00"
1948 DIM MB(7, 2): REM MAX BLOCK PER DEVICE

```





# Readers Data EXchange



```

1950 FM = 768:TM = 771:EF = 772:CT = 773:PD =
      774:CM = 783:UT = 795:BK = 798:HP = 800:DP
      = 806:PB = 826:BU = 855
1960 ID = 63696:LE = 47:PC = 58:RD = 128:WR =
      129:BS = 36864:KB = - 16384:KS = - 16368
1965 NB = 36832:SB = 36834:EB = 36836:FB =
      36838:RS = 36840:S1 = 36848
1968 S9 = 36608: FOR I = 1 TO 7: FOR J = 1 TO
      2:MB(I,J) = 279: NEXT J: NEXT I:MB(5,1) =
      1599: REM SET DEFAULTS FOR MAX BLOCKS
1970 C1 = 4096:C2 = 256:C3 = 16:C5 = 511:C6 =
      255:C7 = 128:C8 = 100:C9 = 10:D$ = CHR$(
      4):I$ = CHR$(9)
1980 SL = 6:DR = 1:HB = 0:HX = 1:HF = 1:HT = 1:VT
      = 0:BL = 0:BF = 0:CC = 0:CF = 0:HE$ =
      "0123456789ABCDEF": GOSUB 1860: GOTO 530
2000 REM SEARCH FOR HEX BYTE PATTERN
2010 VTAB 24: HTAB 27: FLASH: PRINT CHR$(7)
      ">>FIND<<":;ED = 1
2020 SE = 1: VTAB 22: HTAB 1: FLASH: PRINT
      "START": NORMAL: GOSUB 980:B1 = BL
2030 VTAB 22: HTAB 1: FLASH: PRINT "END^":;
      NORMAL: GOSUB 980:B2 = BL
2040 VTAB 22: HTAB 1: PRINT "*****"
2050 POKE 34,0: POKE 35,21: HOME: PRINT "ENTER^
      SEARCH^PATTERN^(16^BYTES^MAX)"
2060 POKE 35,24: HT = 1: VT = 0: OBS = BS: BS = S1: PT
      = 0
2070 OF = 0: BF = 0
2080 IF PT >= 16 THEN 2100
2090 GOSUB 340: GOSUB 600: GOTO 2080
2100 ED = 0: SE = 0: BS = OBS: ONERR GOTO 1080
2110 IF (PT < 1) OR (B1 < 0) OR (B2 < B1) THEN
      2190
2120 POKE NB,PT
2130 POKE SB + 1, INT (B1 / C2)
2140 POKE SB, INT (B1 - (INT (B1 / C2) * C2))
2150 POKE EB + 1, INT (B2 / C2)
2160 POKE EB, INT (B2 - (INT (B2 / C2) * C2))
2170 POKE CM, RD: CALL S9
2180 IF PEEK (RS) = 0 THEN 2230
2190 GOSUB 1920
2200 VTAB 10: HTAB 10: PRINT "BYTE^PATTERN^NOT^
      FOUND": GOSUB 1910: GOSUB 600
2210 BL = 0: BF = 0: PT = 0: VT = 0: HT = 1: POKE
      BU, 144 + BF
2220 GOSUB 550: GOSUB 1060: GOTO 2290
2230 BL = PEEK (BK + 1) * C2 + PEEK (BK)
2240 BF = PEEK (FB + 1): PT = PEEK (FB)
2250 IF (PT + 1) >= PEEK (NB) THEN PT = PT - PEEK
      (NB) + 1: GOTO 2270
2260 IF BF = 1 THEN BF = 0: PT = PT - PEEK (NB) + C2
      + 1
2270 VT = INT (PT / 13): HT = PT - (INT (PT / 13)
      * 13) + 1
2280 POKE BU, 144 + BF: GOSUB 530: GOSUB 550
2290 POP: GOTO 2290

```

## Checksums

```

10 - $BADD 1140 - $696D
20 - $9B13 1150 - $D109
30 - $4D3B 1160 - $6320
40 - $AD92 1170 - $9A33

```

```

50 - $C899 1180 - $BDD1
60 - $FF65 1190 - $5D29
70 - $A3BF 1200 - $E517
80 - $A900 1210 - $302A
82 - $B259 1220 - $AA93
85 - $4A91 1230 - $EF89
90 - $5F00 1240 - $7C93
95 - $376B 1250 - $0FC2
97 - $4816 1260 - $DA45
100 - $076F 1270 - $23D2
110 - $E1F7 1280 - $38D7
120 - $0A5A 1290 - $80D2
130 - $9B45 1300 - $70DA
140 - $0BFA 1310 - $93A2
150 - $A67B 1320 - $A803
160 - $F69A 1330 - $82A2
170 - $F689 1340 - $F2CF
180 - $DF21 1350 - $C1D7
190 - $3348 1360 - $6AA2
200 - $0157 1370 - $FA56
210 - $1DB1 1380 - $F498
220 - $101E 1390 - $D709
230 - $DC80 1400 - $BF4F
240 - $9D60 1410 - $02C5
250 - $53ED 1420 - $512C
260 - $6117 1430 - $F23F
270 - $C0D0 1440 - $98FD
280 - $5750 1450 - $8A3B
290 - $9026 1460 - $ED3A
300 - $A89B 1470 - $1C06
310 - $DF91 1480 - $6448
320 - $F303 1490 - $69D5
330 - $0F50 1500 - $8926
340 - $4898 1510 - $B846
350 - $040C 1520 - $E490
360 - $7961 1530 - $EEC1
370 - $813F 1540 - $1801
380 - $851B 1550 - $B4B8
390 - $E84D 1560 - $25F7
400 - $BB92 1570 - $C245
410 - $DAD2 1580 - $4CD1
420 - $4D4D 1590 - $6E1C
430 - $1930 1600 - $A191
440 - $061A 1610 - $224B
450 - $FA17 1620 - $9A61
460 - $D995 1630 - $45DA
470 - $653B 1640 - $837F
475 - $ED4C 1650 - $C1BE
480 - $61DD 1660 - $03A0
490 - $B926 1670 - $966A
500 - $5458 1680 - $6FE8
510 - $D0EF 1690 - $0D76
520 - $6298 1700 - $EB10
530 - $F971 1710 - $AD52
540 - $BA12 1720 - $B14D
550 - $0EDF 1730 - $44D5
560 - $EB22 1740 - $0EC1
570 - $D4F4 1750 - $D4C3
580 - $DCA7 1760 - $0781
590 - $5482 1770 - $5E30
600 - $8509 1780 - $086C
610 - $CF09 1790 - $B3C5
620 - $52D4 1800 - $AE1A
625 - $F6C2 1810 - $F5D1

```

```

630 - $15A6 1820 - $0AFE
640 - $D439 1830 - $B4C2
650 - $064A 1840 - $73D8
660 - $2E21 1850 - $77B8
670 - $24F3 1860 - $47F7
680 - $12BF 1870 - $1348
690 - $8988 1875 - $FDCF
700 - $15CE 1880 - $AA5C
710 - $EE54 1890 - $7F53
720 - $C19D 1900 - $AFD8
730 - $C88C 1910 - $1DD5
740 - $452A 1920 - $A264
750 - $A575 1930 - $736D
760 - $3D2B 1940 - $939D
770 - $D240 1945 - $ACF8
780 - $394C 1948 - $42E1
790 - $F36A 1950 - $ECE5
800 - $702A 1960 - $2471
810 - $64B0 1965 - $5296
820 - $353C 1968 - $88CC
830 - $1178 1970 - $F339
840 - $EDD1 1980 - $F702
850 - $D75D 2000 - $4D39
860 - $6FBE 2010 - $2638
870 - $C21D 2020 - $85A5
880 - $0F73 2030 - $5CAE
890 - $9AC6 2040 - $E8BC
900 - $CC2B 2050 - $D18A
910 - $680C 2060 - $AB98
920 - $5E66 2070 - $D781
930 - $B1E1 2080 - $604D
940 - $E863 2090 - $8001
950 - $A4DA 2100 - $29A2
960 - $9E9A 2110 - $7429
970 - $9FF9 2120 - $9967
980 - $3280 2130 - $174B
990 - $4FBF 2140 - $A031
1000 - $F627 2150 - $1527
1010 - $9F16 2160 - $E961
1020 - $4874 2170 - $553F
1030 - $2BCE 2180 - $9B3D
1040 - $CBCC 2190 - $0177
1045 - $10C6 2200 - $CCEA
1050 - $DF3B 2210 - $4518
1060 - $FE7D 2220 - $D78D
1070 - $796A 2230 - $3003
1080 - $2C12 2240 - $A3FC
1090 - $8AC0 2250 - $93DF
1100 - $7192 2260 - $44B4
1110 - $1CEC 2270 - $9EFC
1120 - $B001 2280 - $C02F
1130 - $37B8 2290 - $B1CC

```

## OBJ.PROEDIT

```

0300: 01 00 00 00 00 00 20 0C $5C84
0308: 03 D0 08 60 20 00 BF 80 $98FA
0310: 1A 03 60 80 04 03 20 3A $7055
0318: FF 60 03 60 00 90 00 00 $6C0B
0320: AD 03 03 4C 30 03 AD 03 $CEDB
0328: 03 AE 00 03 E0 02 F0 04 $6CF7
0330: 20 DA FD 60 09 80 20 ED $0150
0338: FD 60 A9 01 85 25 20 22 $69F8
0340: FC A9 00 85 24 8D 02 03 $6E63

```



```

0348: A9 0D 8D 01 03 A9 A0 20      $52EC
0350: ED FD AE 02 03 BD 00 90      $087B
0358: AE 00 03 E0 02 F0 0B 20      $9A87
0360: DA FD A9 A0 20 ED FD 4C      $2D51
0368: 8B 03 09 80 C9 A0 10 0E      $C5DC
0370: AE 05 03 E0 00 F0 05 E9      $D1F2
0378: 80 4C 7E 03 A9 AE 20 ED      $B06F
0380: FD A9 A0 20 ED FD A9 A0      $1A00
0388: 20 ED FD EE 02 03 F0 08      $C1BD
0390: CE 01 03 D0 BD 4C 48 03      $7D72
0398: 20 9C FC E6 25 20 22 FC      $3E56
03A0: A9 16 85 22 60                $5858

```

## OBJ.PROED.FIND

```

8F00: A9 FF 8D E8 8F A9 00 8D      $2243
8F08: E1 8F A9 90 8D 2E 8F AD      $FF45
8F10: E2 8F 8D 1E 03 AD E3 8F      $55BF
8F18: 8D 1F 03 20 06 03 AD 04      $5B94
8F20: 03 F0 01 60 AC E1 8F B9      $B349
8F28: F0 8F A2 00 DD 00 90 F0      $4DD3
8F30: 0C C0 00 F0 11 A0 00 B9      $D2D2
8F38: F0 8F 4C 2C 8F C8 CC E0      $6C70
8F40: 8F F0 39 B9 F0 8F E8 D0      $5909
8F48: E3 AD 2E 8F C9 90 D0 0B      $1A69
8F50: 09 01 8D 2E 8F B9 F0 8F      $B40C
8F58: 4C 2C 8F 8C E1 8F A9 90      $B842
8F60: 8D 2E 8F AD 1E 03 CD E4      $E4C9
8F68: 8F AD 1F 03 ED E5 8F B0      $9270
8F70: 1B EE 1E 03 D0 03 EE 1F      $5623
8F78: 03 4C 1B 8F A9 00 8D E8      $DB14
8F80: 8F 8E E6 8F AD 2E 8F 29      $0A32
8F88: 01 8D E7 8F 60                $09AD

```

J. Aufderheide

Softkey for...

## Estimation

Micros for Micros

### Requirements

- Super IOB 1.5
- Sector Editor

*Estimation*, by Micros for Micros, is a great little program to help teach young kids how to estimate. The problem with it, of course, is that it is protected. Now, a protected disk for elementary aged kids is an invitation to disaster, and bit copiers would not touch the disk. What's a person to do? Have you thought of how many ways an elementary aged student can accidentally make a disk so that it never works again?

A boot of the original did not sound quite right. So I booted it on my IIe with a *Trak Star* installed. The program loaded DOS, went to track \$03.5, then went to track \$08. As a matter of fact, the head never read past track \$13. Interesting.

The next step was to break out the CIA and look at track \$08. Both data and address epilogues were FFFF, and the checksum was not quite right. A look at track \$08, sector \$00 showed a VTOC that pointed to a catalog sector at track \$08, sector \$0F. A look at track \$08, sector \$0F showed that all programs seemed to be located from track \$06 to track \$0D. A random check of the rest of the disk showed garbage. Track \$03.5 appeared to be accessed only during boot up. Looked like a nice simple job for *Super IOB*.

I initialized a disk with ProntoDOS, and installed a controller that moved tracks \$06 to track \$0D to my new disk. Putting away the original, as I thought I was done, I then went to work on my copy to make it as normal as possible.

The first job was to break out my sector editor and read in the VTOC from track \$08, sector \$00, change the second byte from \$08 to \$11, and write it back out to track \$11, sector \$00. I read in track \$08, sector \$0F and again changed the second byte to \$11, and wrote the sector to track \$11, sector \$0F. By doing this, I had told their VTOC and catalog that it was relocated on the normal catalog track.

At this point I felt great! Booting the copy. I ran what I thought was the correct program, STARTMENU (seems logical, right?). It did not work. The Hi-Res text did not look like text, but garbage. So I loaded STARTMENU and looked. At line 21 there was a CALL 2048. 2048? That's the normal location for an Applesoft program! Line 12 was HGR, so there was no program at \$2000. That would mean that there was code at \$800 that the program called, and the program must be relocated above Hi-Res Page 1.

A check of memory showed that starting at \$800 we have code that sets up the shape tables that the program uses. Finally, at line 15, the program PEEKed location 254 for value 255.

The next step then was to capture the shape tables, and write a HELLO program that relocates Applesoft programs above Hi-Res Page 1 and puts 255 into location 254 (under the assumption that that location is checked by other parts of the program!).

So dig out that original again! Booting the original disk, I jumped into the monitor, and moved the machine code to \$2000 where it would be safe (2000<800.1999M) and booted my copy. I then re-entered the monitor (CALL -151), moved the code back where it belonged (800<2000.3199M) and saved it on the copy (BSAVE SHAPES, A\$800, L\$1199). Then I wrote the HELLO program at the end of this article which relocates Applesoft programs above Hi-Res Page 1, sets location 254 to 255, loads SHAPES at \$800, and runs the program STARTMENU, and saved it on my copy (SAVE HELLO).

A quick boot, and it worked! I now had a safe copy of *Estimation*, and need no longer worry about bombing my original. For those of you in a hurry, here's the cookbook:

**1** Initialize a disk with a fast DOS.

### INIT HELLO

**2** Install the controller at the end of this article and copy tracks \$06 to \$0D to your copy.

**3** Break out your sector editor and read track \$08, sector \$00 from your copy. Change the second byte from \$08 to \$11. Write the sector out to track \$11, sector \$00 or your copy.

**4** Read track \$08, sector \$0F from your copy. Change the second byte from \$08 to \$11. Write the sector out to track \$11, sector \$0F.

**5** Boot your original and break into the monitor. Move the code to a safe place:

2000<800.1999M

Now boot your copy and move the code back:

800<2000.3199M

Save it on your copy:

BSAVE SHAPES, A\$800, L\$1199

**6** Enter the Applesoft program at the end of this article and save it on your copy:

SAVE HELLO

**7** Put your original away, and enjoy your copy!

## Controller

```

1000 REM ESTIMATION CONTROLLER
1010 TK = 6:LT = 14:ST = 15:LS = 15:CD = WR:FAST = 1
1020 POKE 47426, 24
1030 GOSUB 490: RESTORE: GOSUB 170: GOSUB 610
1040 GOSUB 490: GOSUB 230: GOSUB 610: IF PEEK (TRK) = LT THEN 1060
1050 TK = PEEK (TRK):ST = PEEK (SCT):GOTO 1030
1060 HOME: PRINT "COPY DONE": END
1070 DATA 255, 255, 255, 255

```

## Controller Checksums

1000 - \$356B	1040 - \$D0D0
1010 - \$0C00	1050 - \$C320
1020 - \$331B	1060 - \$8C73
1030 - \$8DF8	1070 - \$8246

## HELLO

```

10 IF PEEK (103) + PEEK (104) * 256 <> 16385
   THEN POKE 103, 1: POKE 104, 64: POKE 16384, 0: PRINT CHR$(4) "RUN HELLO"
20 POKE 254, 255: PRINT CHR$(4) "BLOAD SHAPES, A$800": PRINT CHR$(4) "RUN STARTMENU"

```





## Zorro

Question: I have a question concerning John Wiegley's article about using the cassette jacks for 'Better Sound.' How can you get better sound if you can barely hear anything from your headphones?

Somebody, anybody, PLEASE send in a schematic diagram for an amplifier circuit!!

## Jack R. Nissel

### Softkey for...

## Perry Mason The Case Of The Mandarin Murder

Telarium

**1** Use any fast copy program to copy all four sides.

**2** Sector edit side one of the copy you just made.

Trk	Sct	Byte(s)	From	To
\$13	\$0B	\$5A-5C	AD 82 C0	20 29 1C

### Softkey for...

## The Hobbit

Addison-Wesley Publishing Company

The softkeys in COMPUTIST #34 and COMPUTIST #35 did not work on my copy. Neither side could be copied using a fast copy program, including COPYA as suggested in the softkeys. Modifying COPYA, however, did work.

**1** Boot your DOS 3.3 system disk.

**2** Tell DOS to ignore errors and use COPYA to copy both sides of the original disk.

**POKE 47426,24**  
**RUN COPYA**

**3** Make the following sector edits to side 1 of the copy.

Trk	Sct	Byte(s)	From	To
\$00	\$01	\$2A-2C	4C 2A 02	EA EA EA

### Softkey for...

## Castle Wolfenstein

Muse Software

I was not able to deprotect this using any of the softkeys in COMPUTIST. If you had the same problem this softkey should work.

**1** Boot your DOS 3.3 system disk

**2** Put a blank disk in the drive and enter:

FP

INIT ^HELLO,V1

DELETE ^HELLO

The V1 will give it a volume number of 001 which is the same volume number as the original. Put this disk aside, you will use it later.

**3** Boot the original *Castle Wolfenstein II* disk and after your drive reads the first 3 tracks reset into the monitor.

**4** Move the RWTS to a safe location:

1900<B800.BFFFM

**5** Put your slave disk in the drive and boot it.

**6** Insert your IOB disk and save the RWTS:

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

**7** Install the controller listed below into *Super IOB* and copy the original to the slave disk. When asked, enter N, so as not to format the disk while running the controller.

### Controller

```
1000 REM CASTLE WOLFENSTEIN CONTROLLER
1010 TK = 3: ST = 0: LT = 35: CD = WR
1020 T1 = TK: GOSUB 490: GOSUB 360: ONERR GOTO 550
1030 GOSUB 430: GOSUB 100: ST = ST + 1: IF ST < DOS
    THEN 1030
1040 IF BF THEN 1060
1050 ST = 0: TK = TK + 1: IF TK < LT THEN 1030
1060 GOSUB 490: TK = T1: ST = 0: GOSUB 360
1070 GOSUB 430: GOSUB 100: ST = ST + 1: IF ST < DOS
    THEN 1070
1080 ST = 0: TK = TK + 1: IF BF = 0 AND TK < LT THEN
    1070
1090 IF TK < LT THEN 1020
1100 HOME: PRINT "COPY DONE": END
10010 IF PEEK(6400) < > 162 THEN PRINT CHR$
    (4) "BLOOD RWTS.CASTLE.A$1900"
```

### Controller Checksums

1000	- \$356B	1060	- \$90D8
1010	- \$3565	1070	- \$98DD
1020	- \$E1E8	1080	- \$7422
1030	- \$F7E9	1090	- \$2DB2
1040	- \$D35A	1100	- \$1A55
1050	- \$E5B5	10010	- \$7F32

You can put fast DOS on this copy if you want to.

NOTE: The original disk is a 13 sector disk, so when the controller reads it, it will "BEEP" three times on each track. This is fine, the copy will be OK.

### Iigs Softkey for...

## Leisure Suit Larry in the Land of the Lounge Lizards

Sierra On-Line

**1** Use any fast copy program that can ignore errors and copy both disks.

**2** Edit side one of the copy.

Block	Byte(s)	From	To
\$0A5	\$159	22	AF

### For Your Information

If you have the *Print Shop Iigs* try the following.

Boot your disk, and when you get to the first screen after the title screen move down to "Exit Program". Hold down the OPEN-APPLE key and double click your mouse to exit the program.

The second thing to try is to change the date in the control panel to December 25. Save the change and then boot your disk. Look at the first screen after the title screen, up at the top where it usually says "The Print Shop". I won't tell you what will happen in either case, but I think you will like the surprises that Broderbund put in the program. If anyone knows any others, please send them in.

In response to the letter from Bud Meyers that appeared in COMPUTIST #52. First, I am sorry that he seems to have had so much trouble in getting the softkeys to work for his software, but to say that the instructions leave "a great deal to be desired", especially in light of the fact that he says that he is somewhat knowledgeable in computers, makes no sense to me. When I first subscribed to COMPUTIST, about 11 months ago, I was a complete novice as far as computers go. Since then, I have gotten all of the issues going back to COMPUTIST #1. I can honestly state that I have yet to see an issue of COMPUTIST that has not had a softkey that would not work for at least one title that I have, either as the softkey was written, or with me making some minor modification to it, such as looking for the bytes to edit at another spot on the disk, changing a line in the Super IOB controller, etc.

He also states that he doesn't care if his copy retains the protection. Again, if he is at least somewhat knowledgeable, he must realize the advantages a deprotected copy offers over one that is protected. I cannot believe that Mr. Meyers could be content with the bit copy programs if he is unhappy with the softkeys. I would say that my failure rate in using a softkey on the title that it was made for is less than 5 percent. I know of no copy program that



can come close to matching this and even if they could the resulting copy would still be protected.

Rather than end on a critical note, I would like to say that if Mr. Meyers would care to send in a list of the software that he has had trouble in deprotecting, I am sure that one or more of the COMPUTIST readers, including myself, will help him if we can.

John R. Pierce

## Putting *Flashcalc* on an 800K Disk

### ■ Requirements

- Any Apple
- Ramfactor(AE)
- Sector editor
- Flashcalc* program disk

In COMPUTIST #23, Doni Grande told us how to deprotect *Flashcalc*. Since then the 800K disk has arrived and I wanted both program and data on one disk. I like the program because it is quick and dirty for simple tasks.

Alas, *Flashcalc* would not allow you to install it on the 800K disk. *Flashcalc* does allow you to install the program on the hard disk. Since a ram disk is just like a hard disk, I tried installing the program on the Ramworks card in my IIe. Again, the program would not install itself. I next tried the Ramfactor. To my surprise, the program installed itself on the card. All I had to do was transfer the files to an 800K disk and I would be ready. Wrong. Even though the programs that were needed to run *Flashcalc*, FC.SYSTEM AND FC.PROG., were on the 800K disk, the program would not go.

Luckily, I had COMPUTIST #23. As I studied the info from the article, I was able to find how to make *Flashcalc* run from the 800K disk.

### Cookbook

**1** Follow instructions and install *Flashcalc* on the Ramfactor or similar card.

**2** Transfer the files FC.SYSTEM and FC.PROGRAM to a directory on the 800K disk. (I called my disk /FLASH and the directory /FLASHCALC.)

**3** To make the sector edits easier, copy the file FC.SYSTEM to a 5-1/4" blank disk. The file occupies two sectors.

**4** Using a sector editor you must change a value in one of the sectors (I used a DOS 3.3 sector editor) and the pathname in the other.

**5** If you put FC.SYSTEM on a blank disk, perform the following sector edit:

Trk	Sct	Byte(s)	From	To
\$00	\$01	\$FE	18	38

**6** In the second sector track \$00, sector \$0F you must change the pathname to that on your 800K disk. This starts at byte \$B1. The byte at \$B1 is the value for the length of the pathname. If this value is wrong, the program will not run.

Also, do not worry about overwriting the values after \$B1. These bytes are reserved for the pathname. The programmer put data there to fool you. All bytes must be in HEX. (F=46, L=4C, A=41, etc.)

**7** Example of entering pathname:  
pathname: /FLASH/FLASHCALC/  
# of letters = 16 (Do not count the final '/' as part of the length.)

HEX value to put at \$B1 = 10.

The complete entry would start at \$B1 and look like: 10 2F 46 4C 41 53 48 2F 46 4C 41 53 48 43 41 4C 43 2F 00. After the last '/', enter 00 to mark the end.

**8** Write the sector back to the 5-1/4" disk.

**9** Copy the file FC.SYSTEM back to the 800K disk into the proper directory.

The FC.SYSTEM file will now run *Flashcalc*. You can put *Flashcalc* in whatever directory you want as long as the correct pathname is in FC.SYSTEM.

I wrote Paladin Software several times to see if they would help. There was no answer to my letters. Where have all the software companies gone?

Stanley Planton

## A Copy-protection Scheme for ProDOS

### ■ Requirements

- EDD 4.N
- COPY II Plus ProDOS, or similar utilities
- Sector editor
- Blank Disks
- A ProDOS file to be protected

One result of my article in COMPUTIST #51 was a sarcastic remark in another publication, to the effect that I'm out of step with the norm. Everyone else is interested in the REMOVAL of copy protection, while I seemed to be advocating the creation of new schemes. Oh, well, I guess they missed the

"know your enemy" point of the exercise. Actually, sometimes the protection scheme is more interesting than the program it protects!

There is a request on page 33 of COMPUTIST #52 from a Mr. David Erickson for a way to protect in ProDOS. While I can't implement his preferred epilogue check, I have a way to copy protect ProDOS files, based on my COMPUTIST #51 article, again using EDD 4.N to do the real work.

A brief explanation of the relationship between DOS and ProDOS seems to be called for by way of introduction. When the gang at Apple designed ProDOS, they did not start with a clean sheet of paper. ProDOS seems to be a refinement of the SOS that they developed for the Apple III, which in turn was an evolution of DOS 3.3.

While there are many differences between DOS 3.3 and ProDOS, there are even more similarities. The two systems share general track and sector formats, and run in the same drives. In fact, there are many apparent similarities in the way that they access and store some of the address and data identification bytes found on the disk. This latter similarity can give us a way to protect ProDOS files that is virtually identical to the protection of DOS 3.3 files I explained in COMPUTIST #51. We can make the address headers on the disk vary from D5 AA 96 to another value, such as D4 AA 96, without a great deal of effort. This will "shut down" such popular copying programs as *Locksmith Fast Copy*, *Disk Muncher*, and *COPYA*, since they can no longer determine the start of a sector on a track.

Most of us are used to entering the monitor after booting a DOS 3.3 disk, and making changes to memory locations in order to eliminate error-trapping or byte-recognition routines. What should be realized is that the original values are loaded from the disk, and we can make the same changes to the DOS on the disk. For example, if we wish to tell DOS to ignore the first byte of an address header, we can enter the monitor, change locations SB954-B955 to 29 00 and return to BASIC with the task accomplished. We could also search the disk for the string 10 FB 09 D5 and replace the 09 D5 with 29 00, accomplishing the same end. When this disk is booted, the DOS loaded would ignore the first byte of the header.

ProDOS uses the same bytes to accomplish the same thing! In fact, there are many parallels between the ways the two operating systems read individual sectors.

To protect a ProDOS disk from being easily copied, follow these steps:

**1** Format a ProDOS disk and copy ProDOS to the disk.

Most versions of ProDOS will use tracks \$01-\$04 if they are the first file on the disk, so let's keep ProDOS there.





**2** Copy the file(s) to be protected to this disk. Get a map of the disk, using *COPY II Plus*, in order to verify that the next file(s) will begin using tracks \$05-\$22. These will be the tracks we'll protect with EDD.

**3** Write protect this disk. EDD likes to have the "original" protected.

**4** Boot up EDD 4.N, and place the disk you just write-protected into drive #2, and a blank into drive #1.

**5** Copy tracks \$00 - 04 with NORMAL settings. Note that EDD doesn't use hexadecimal in its track numbering!

*The following is a reprise of my COMPUTIST #51 approach:*

*There is a trick to entering values into these EDD routines: you must use the spacebar between MOST entries; only use the **RETURN** following the value entered into the #00 instruction and the final value. I know that this sounds unclear, but try it.*

*Remember, in COMPUTIST the required **SPACE** is indicated by ^ and **RETURN** must be pressed after the command line.*

**6** From the main menu of EDD, select the "change parameters" option.

**7** From the next menu, select the "reprogram prewrite" option.

**8** Enter the following:

```
00 = 20
20 ^ 31 ^ D5 ^ AA ^ 96 ^ 36 ^ D4 ^ AA ^ 96 ^ 21 ^ 10
Q                                     to quit
```

Now go back to the main menu of EDD and copy the rest of the tracks (\$05-34). EDD should produce tracks that have D4 AA 96 for headers, which is what we want.

**9** Try to boot the disk. It should boot, but you should also get an error message when ProDOS tries to read the first "SYSTEM" file. You did remember that ProDOS requires a "SYSTEM" file, didn't you?

**10** Scan the disk for 10 FB C9 D5 and change the C9 D5 to 29 00.

I use the sector editor of *COPY II Plus* to do this, but many other programs will do. Remember to write the changes back to the disk! For many versions of ProDOS, the bytes will be located in track \$04, sector \$0A, starting around address \$A8 (there may be some variation depending on which of the incarnations of ProDOS you're using).

You should now have a disk with normal catalog and ProDOS tracks, but which has a degree of copy protection.

I'd like to pose a challenge/question to others: just how similar ARE these routines that DOS 3.3 and ProDOS seem to share? Are there any other tricks that can be easily applied to both?

### Softkey for...

## Random House Media

Volume 1 and Volume 2

This software includes most packages sold by the Random House School Division that are DOS 3.3 or ProDOS based, including the following titles:

- The Boars' Store
- The Boars Tell Time
- Animal Alphabets and Other Things
- Inside Outside Shapes
- Inside Outside Opposites
- City Country Opposites
- Snoopy's Reading Machine
- Snoopy's Skywriter Scrambler
- Charlie Brown's ABCs
- Typing is a Ball, Charlie Brown
- Charlie Brown's 123s
- Tutorial Comprehension (30 disks of reading comprehension programs, titles will vary, but include the words "Tutorial Comprehension")
- Word Blaster (2 disks)
- Homonyms in Context
- Word Mount
- Word Count
- Customized Alphabet Drill
- Alphabet Sequencing
- Snoopy Writer
- Story Starter
- Story Builder
- Spelling Demons
- Fundamental Spelling Words in Context...
- Fundamental Capitalization...
- Focusing on Language Arts (9 disks, titles will vary, but include the words "Focusing on Language Arts")
- Customized Flash Spelling
- Fundamental Punctuation Practice
- Mathematics Series (23 disks, titles will vary, but start with the word "Mathematics" in small type)
- Individualized Study Master
- Career Focus
- Random House Library Management Programs (7 disks, titles will vary)
- Peanuts Math Matcher
- Mr. and Mrs. Potato Head
- Snoopy to the Rescue
- Peanuts Picture Puzzlers
- Peanuts Maze Marathon
- Stock Market Simulation
- Fix It
- Tournament Bridge

### Requirements

- System Master
- Copy II+
- ProDOS
- Speed DOS
- Some knowledge of BASIC helpful
- Blank disks, disk notcher, and time

Advanced users: Go to the end of the instructions.

Beginners: Follow through step-by-step.

A friend of mine, who is a teacher, recently left two huge three-ring binders of Random House software with me overnight. These appeared to contain virtually the entire current catalog of the publisher! He wanted to know how he could make backups of the programs. The software is copy-protected, so I plowed into the disks in the search of truth, beauty, and an all-purpose softkey. There were a LOT of disks to analyze and not much time, so the following may not be "complete" for every program. The following procedures should, however, provide a backup of most or all of the educational software in question.

Random House seems to publish a wide range of ostensibly "educational" software. There is, however, an equally wide range of educational quality involved. Many of the programs are cute and funny, such as the *Peanuts* series, but of little educational value in the classroom. Others, such as *Fix It*, provide a real challenge to higher-order thinking skills.

The tutorial and management programs seem to fall in between. When the software is converted to standard DOS and ProDOS files, there is an apparent varying in the quality of the programming, as well. Very nice graphics can be loaded by fairly "clunky" BASIC routines, and most of the DOS-based software seems to have been constructed from a core of Applesoft, which explains its leisurely pace of operation. Many of the programs are irritating to run repeatedly, since a LOT of time is wasted in running Random House commercials. Since many of these are BASIC routines that call up graphics screens, a good programmer should be able to speed up the loading of the program by re-routing the software to skip the commercials.

The first step in analyzing the software is to look at the disk label. If there is a small "APPLE DOS 3.3" statement, you can assume that the program is DOS-based. Otherwise, assume that it is ProDOS. If in real doubt, boot the original and watch for a ProDOS copyright screen!

I used the *Nibble Editor* from the bit copy menu of *COPY II Plus* to examine tracks on the Random House disks. To do this, boot up *COPY II Plus*, select "Bit Copy" from the main menu, tell the program what slot your drive controller is in (usually 6), and select "NIBBLE EDITOR" from the next menu.



You can now remove the *Copy II Plus* disk and insert the disk to be examined. Pressing "Q" will allow you to examine track after track, while pressing "A" will analyze the track currently being examined.

An unprotected DOS 3.3 or ProDOS disk will display a pattern of bytes when analyzed this way, and the pattern will usually start with a series of highlighted FF bytes, called "sync bytes". Ignore these, but look at the next series of bytes.

There may be a stray byte or two, then a pattern of D5 AA 96, called the "address header", used by DOS as a kind of wake-up call. Next will be a series of bytes that contains the information for DOS to figure out where on the disk the read-write head is currently located (i.e. the address of the sector), and a series of "address epilogue" bytes, DE AA.

There may be more stray values and some FF sync bytes, then the bytes that identify the start of the data within the sector (the "data header" bytes): D5 AA AD. Pressing the "M" key will allow you to page down in the track to examine the "data epilogue" bytes, DE AA.

Note the pattern: information STARTS with D5 AA — (a 96 or an AD following), and information ENDS with a DE AA pattern.

On the Random House disks, the publisher has modified this standard system to copy-protect the disks: the "header" bytes, D5 AA 96 and D5 AA AD are normal, but the "epilogue" bytes have been changed in many cases. On most of these disks, epilogue bytes will read AA DE or EB AB when analyzed. This is a fairly minor change, and can be compensated for by telling a standard DOS to totally ignore the epilogue bytes when it reads a disk.

A good tool for unprotecting disks like these is COPYA, which is greatly misunderstood and unappreciated by the non-hacker community.

COPYA will format a disk in drive #2 under standard DOS 3.3 (which includes ProDOS formatting as well). If we can persuade COPYA to ignore errors such as altered epilogue bytes on the ORIGINAL disk, it will handily copy the contents of the copy-protected sectors to the disk that it just formatted under standard rules; it will UNPROTECT the program. Note that it does NOT copy the protected tracks "as is"; it simply TRANSFERS data from protected tracks to unprotected tracks, a much more useful activity!

Now that we know how Random House copy protects its disks, we can boot up our System Master disk, tell DOS to ignore weird epilogue bytes, and RUN COPYA. Since many of the Random House disks are double-sided, have a disk notcher ready, and remember to copy both sides!

**1** Boot the System Master.

**2** Enter the monitor.

### CALL-151

**3** Change DOS to ignore the epilogues then return to BASIC.

**B988:18 60**  
**B925:18 60**  
**3DOG**

**4** If you have a DOS 3.3 based Random House disk, you should be able to CATALOG it at this point. Usually, the "boot program" will be the first AppleSoft program in the catalog, and may be named HI, START, HELLO, SIDEA, or something similar.

**5** Quit playing and put the System Master disk back in the drive before you destroy what you've done in step #3.

### RUN COPYA

**6** Put the Random House disk in drive #1, and a blank (notched for copying side 2, if necessary) into drive #2, and press RETURN until COPYA starts.

**7** When prompted to copy another, answer "Y" and flip both disks (if the original is double-sided), and repeat.

**8** CATALOG the resulting product with a ProDOS version of *COPY II Plus*. If the disk returns a DOS 3.3, type of catalog, copy a fast DOS to the disk AFTER determining the name of its "boot" program. Change the boot program name to match the original boot program. The reason for this operation is that DOS keeps the name of the boot program within its own tracks; when you change DOS on a disk, you should also put the name of the correct boot program into DOS.

**9** If the Random House disk returns a ProDOS-type of catalog, DELETE the ProDOS from the copy you made, and copy a version of ProDOS from a standard ProDOS disk. Note that different versions of ProDOS take up different amounts of room on the disk, ranging from 30 to 32 "blocks". On most of these disks, there is room for the larger and more recent versions of ProDOS.

**10** Remember to check the flip side of the copy. Some of these disks are "bootable" on both sides, and you'll have to put either DOS or ProDOS on that side, as well, if you want the disk to work.

**11** Make a second copy of the disk, and try to analyze the BASIC "boot" program so you can modify it to skip the commercials for Random House, Pelican, the programmers and all their friends, etc., and boot instead into the "real" program!

Kim Griffith

Softkey for...

## Master Diagnostics II & II+

Nikrom Technical Products

### Requirements

- Apple II, II+
- FID
- blank disk

After seeing the softkey for *Master Diagnostics IIc* in COMPUTIST #51, I remembered I had not been able to make back up copies for *Master Diagnostics II, II+*. The original disk does allow for one backup copy to be made and attempts to copy with the popular copy programs were fruitless.

First, *Locksmith's Fast Copy* routine was used to see if any sectors were readable by normal DOS. None were, so a nibble dump of several tracks showed altered field markers. Instead of trying to figure out what needed to be POKE'd into memory, capturing RWTS is easier.

Here's how to make a COPYA-able disk:

**1** Initialize a blank disk with normal DOS 3.3 and use HELLO as the boot program name.

### INIT HELLO

**2** Delete the HELLO program.

### DELETE HELLO

**3** Boot *Master Diagnostics*. Before the menu appears, press **RESET** several times.

**4** Enter the monitor.

### CALL-151

If you cannot enter the monitor, go back to step 3 and try again.

**5** Once you are in the monitor, move the current RWTS to a safe place in memory.

**8600<B600.BFFF**

**6** Insert the disk you just initialized and load normal DOS 3.3.

**C600G**

**7** Re-enter the monitor.

### CALL-151

**8** Move the RWTS back to its normal place.

**B600<8600.8FFF**

**9** Go back to BASIC.

**3DOG**



**8** Move the RWTS back to its normal place.

**B600<8600.8FFFM**

**9** Go back to BASIC.

**3D0G**

**10** Insert a disk with FID and copy all files except the first two.

**BRUN FID**

You now have a copy that is in normal DOS 3.3 format.

Some of you might have noticed the protected RWTS is also being used to write the files to a normal DOS 3.3 disk. You're right. The protected RWTS works with normal DOS 3.3, but not vice versa. I assume this is because the protected RWTS is told to ignore the non-standard markers.

*Softkey for...*

## Studio II

Syntech AKA Music Digital

### ■ Requirements

- Apple II+, IIe
- Sector editor
- Any DOS 3.3 disk with at least 106 free sectors

*Studio II* is a MIDI sequencing program from Syntech. Apparently, Third Street Software wrote the program, as they take credit on the screen. The program was expensive and no provision was made for making a back up. Repeated attempts with several popular copy programs proved fruitless. Letters to Syntech were never answered and eventually the price of the program dropped, leading me to believe Syntech is now Defuntech, and making the need for a backup even greater.

Copying with *Locksmith Fast Copy* showed several tracks of unreadable sectors. Under DOS 3.3 the disk could be cataloged, showing one binary file, BSEQ. BLOADing the file gave the popular I/O error. Using a sector editor under DOS 3.3 showed two files in the catalog, BSEQ and another binary file with the name consisting of six Control-Hs. Checking for the boot program name showed the six Control-Hs was indeed the boot program.

Control-H is the character for backspace, that's why it didn't show when the disk was cataloged. You can't BLOAD the file as is because six Control-Hs from the keyboard will just backspace you over the BLOAD command. Changing the name in the catalog sector to HELLO made it possible to BLOAD the greeting program. Checking the data sector for the BLOAD length and location of the greeting program showed it loaded at \$200 to \$3BF.

\$200 is the keyboard buffer which means there could easily be a problem, which is certainly what was intended. Loading the greeting program at \$1200 allowed it to be disassembled and traced.

The program skips around a bit, another attempt to confuse us, and makes modifications to DOS, then does some File Manager calls and then normalizes DOS. References to the strobe data latch indicated some timing critical data. Further tracing, the code showed a branch instruction, with one branch going to the UNABLE TO LOAD PROGRAM message, the other branch going to a location that does a JMP \$1000, a likely place for a program to start. Substituting a RTS for the JMP just might put us in the monitor instead of giving control to the program. Because of the calls to the File Manager and the jump to \$1000, it would be a good guess that the BSEQ program is loaded in before the jump takes place.

We can't run the greeting program at \$1200 because BSEQ would overwrite it before it was done executing. Better to run it at the location where it was intended to be run.

What we can do is BLOAD the greeting program at \$1200, make the necessary modification, and enter a short program that will move the modified greeting program down to \$200 and execute it. This will run the greeting program, load the BSEQ program and return to the monitor. This still leaves something awry in DOS, but booting a DOS 3.3 disk with no greeting program normalizes DOS and allows the sequencing program to be BSAVED. Once the program is saved it can be BRUN as any normal binary file.

**1** Copy the original disk with any quick copy program and ignore errors.

**2** With a sector editor, change the name of the greeting program on your copy disk from six Control-Hs to HELLO. The sector that contains this file listing should be track \$11, sector \$0F.

**3** Boot the copy disk.

**4** When the drive stops, load the greeting program.

**BLOAD HELLO, A\$1200**

**5** Enter the monitor.

**CALL-151**

**6** Change the JMP to a RTS.  
**124A:60**

**7** Enter this code to move the modified greeting program to its normal location.

```
F00:A0 00 B9 00 12 99 00 02
F08:C8 D0 F7 B9 00 13 99 00
F10:03 C8 C0 C0 D0 F5 4C 00
F18:02
```

**8** Install your original disk.

**9** Execute the code.

**FOOG**

**10** Install any DOS 3.3 disk with at least 106 free sectors and save the program.

**BSAVE BSEQ, A\$1000, L\$6800**

*Playing Tips for...*

## Under Fire

Avalon Hill

If the time consuming disk access of *Under Fire* is more than you want to put up with, you can use Beagle Brothers' Pronto DOS. The boot process is time consuming because eight binary files are loaded in a FOR/NEXT loop. These files are consecutive in memory so they can all be loaded in memory at one time and saved as one file. This eliminates seven accesses to the disk. Two other files can be made as one, eliminating one more access. If you don't care to see the Hi-Res picture at the beginning, the access to that file can be eliminated, too. The original greeting program is a good example of how not to program in BASIC, you might want to rewrite it completely.

The following modifications were done on an early version of *Under Fire*, later versions may be better. Check to make sure your version matches the information below before you make any changes to your disk.

With a backup of your cracked copy, do the following while in BASIC:

**1** Load the following files at the indicated locations:

```
BLOAD ST0, A$1400
BLOAD ST1, A$1480
BLOAD ST2, A$1500
BLOAD ST3, A$1580
BLOAD ST4, A$1600
BLOAD ST5, A$1680
BLOAD ST6, A$1700
BLOAD ST7, A$1780
BLOAD UFG
BLOAD UFD
```

**2** Now save the files:

```
BSAVE STFILES, A$1400, L$3D1
BSAVE UFFILES, A$4000, L$28DD
```

**3** Load the HELLO file:

**LOAD HELLO**

**4** Delete some unneeded lines.

**DEL 45,50**

**5** Change some other stuff:

```
30 PRINT CHR$(4) "BLOAD STFILES"
40 PRINT CHR$(4) "BRUN UFFILES"
```



**6** If you want to eliminate the loading of the Hi-Res picture, do this. Otherwise, skip it and continue.

**2 POKE 49236,0**

**7** Save the new greeting program.

**SAVE HELLO**

**8** Now get out your sector editor, boot it up, insert your copy disk and locate the first DATA sector of STFILES. To make sure you have the right one, the first four bytes should be 00 14 D1 03. Change the second byte to 04 and write the sector back to disk.

What you have done is to load the files from the loop into a different place in memory because they would overwrite the text screen. The files were saved as one at the new place in memory, and a simple sector edit changed that location so DOS will load them back where they should be. This modification was done on an early version, later versions could be different.

† *Softkey for...*

## Under Fire

Avalon Hill

In COMPUTIST #52, Donald Jones mentioned he had a problem with the *Under Fire* softkey. With my version, it was necessary to change location \$649A to \$60 (RTS). This stopped the disk from initializing itself if it didn't find what it was looking for.

Does anybody know where I can obtain a copy of "What's Where In The Apple" by William Lubert, from Micro Ink (who also doesn't answer their mail)?

**Jim Hart**

COMPUTIST #53 just came in and I have a few comments on it. First of all, the format is really great and I think a lot more people are getting useful stuff out of it. Now, if a particular softkey doesn't work for you, there are probably 3-4 other softkeys for the same piece of software. One of them might just work on your version.

Thanks go out to Brian A. Troha for his long and in depth IIgs softkeys. The disassemblies are helping me to learn 16 bit assembly language.

**To Les B. Minaker:** you might want to write Don Lancaster (Ask the Guru - Computer Shopper) and see if he has a solution for your 128K RAM board problem.

Don's address is:

Don Lancaster  
Synergetics  
Box 809-G  
Thatcher, AZ 85552

**To Chris Wood:** if you have *Bag of Tricks*, use INIT to re-skew your deprotected *California Games* to a 9 descending skew. I think that should speed it up a bit.

**To Stephen Lau:** the *Tower of Myraglen* softkey was great. Now, could you present a short disassembly of the protection code and what it does? It would help us all out. A friend of mine thanks you for the *Dream Zone* softkey.

**To Sandford Eubanks:** thank you for the *Lords of Conquest* softkey. I have been working on that darn program for a month now (off and on). I found the first protection easily, but the encoded one was giving me fits.

**To Robert Ashton:** I agree with you about putting the IBM-Rdex section in the middle of the magazine so that users of either type can disregard the other part. I personally am interested in reading the IBM softkeys, even though I will never buy an MS-DOS machine.

Andrew Swartz's review of *Copy II Plus* v8.1 (in COMPUTIST #53) was accurate, but I have a few complaints about the program itself. First of all, if you are copying a 3.5" disk and you forget to format or insert the backup disk, then the first read of the original is wasted. This is a problem when you have a one megabyte RAM card and the entire disk has just been read in. It does not allow you to "try again" with another disk, nor does it ask you if you would like to format the disk. Now, one of the copy options is COPY W/FORMAT, but I'm talking about the regular copy disk function. My next problem is with the sector editor. The 5.25" and 3.5" bit copiers have different sector editors. I personally like the 3.5" sector/block editor. I would like it if the sector/block editor had all of the functions that the one in *Copy II Plus* v5.5 did, i.e. a follow file function. For large ProDOS volumes, this would be invaluable. I don't prefer the sector/block editor because of this deficiency; instead, I prefer *Block Warden* from the ProSEL package. It has a follow file function, among other things. Not enough has been said about ProSEL, but let me say that it is worth the \$40 that it retails for. I don't see how you could have a hard disk and be without it.

A friend of mine runs a BBS here in town and if you would like to call, the info is:

Greg's Grapevine  
(919)-324-2048  
300/1200/2400 24 hrs  
Uses Apple IIgs (running under GBBS PRO)  
Supports Commodore & Apple

I am on the board if you want to exchange E-mail. By the way, if anyone out there wants to talk "COMPUTIST" talk (via mail), my address is:

Jim S. Hart  
311 Bordeaux St.  
Jacksonville, NC 28540

*Softkey for...*

## Coveted Mirror

Polarware

### Requirements

- COPYA or equivalent
- Coveted Mirror* original
- A blank disk
- Sector editor/disk searcher

The version of *Coveted Mirror* that I have is ProDOS based and uses the Comprehend feature, which allows full & multiple sentence commands. I used the softkey for *Crimson Crown* (COMPUTIST #44, pp. 25-26) with a little modification to deprotect this one.

The first protection routine is loaded in \$4600 for *Coveted Mirror*, unlike \$3F00 in *Crimson Crown*.

**1** Copy the disk with any whole disk copier.

**2** Search the disk for the byte sequence: A0 89 A9 and replace it with: 4C A3 46.

**3** Search the disk for the byte sequence: 4A 6D A7 4A and replace it with: 46 B9 8F 1A.

You're done!

*Softkey for...*

## Rocky's Boots v.4

Learning Company

### Requirements

- Rocky's Boots* v.4 original
- A blank disk
- COPYA or equivalent
- Disk searcher/sector editor

**1** Boot up DOS 3.3 and disable error checking:

**POKE 47426,24**

**2** Copy the original disk:

**RUN COPYA**

**3** Make the following sector edits:

Trk	Sct	Byte(s)	From	To
\$00	\$01	\$1B-1D	BD 8C C0	D0 73 EA

You're done!





Jeff Hurlburt

## The PRODUCT MONITOR

### Ratings

★★★★★	SUPERB
★★★★	EXCELLENT
★★★	VERY GOOD
★★	GOOD
★	FAIR
☹	POOR
☹☹	BAD
☹☹☹	DEFECTIVE

### Stick Wars '88

While the advent of the graphics tablet, trackball, and mouse have greatly impacted its scope of application, the joystick remains the preferred input device for action games, many simulations, and even a few utilities. Picking a good one is important. Since, sooner or later, every Apple and IBM user is 'in the market', this month's column begins with a look at several top contenders in the joystick sweepstakes.

Admittedly, a superficial sampling of today's joysticks might well lead one to conclude that one stick is as good as another. All of the units reviewed here offer handle-mounted fire buttons (i.e. "P0" in Apple parlance), and, with only two exceptions, employ slotted-band drive mechanisms. Gone are such exotic innovations as pressure sensor resistance elements, all the sticks use dual right-angle-mounted pots, and magnetic centering, spring-loaded centering is universal. In fact, none of the units reviewed exhibits the sort of 'feel' roughness, gross centering 'thump', or other usual disqualifiers that used to make it easy to pick one or two clear winners.

Given six viable alternatives, how then do you zero-in on the right stuff? My approach was to first install each joystick and live with it for a week or so. After this 'stick of the week' phase, I gathered all the sticks and ran each through quickie sessions with *Elite* (a good 'action game' test) and *Blazing Paddles* (to

check graphics controller type responses). Finally, using a simple program to provide continuous readout of X,Y values and button status, I made several objective measurements.

The sticks are different. For starters, only Premium III and Flight Stick put short-travel, snap-action ("SA") fire buttons in the handle. Since these require less finger pressure and let the user feel a distinct 'click' when activated, the payoff is better speed and less fatigue in high-action games, and, for graphics work, less chance of messing-up a drawing. Mach III's large, recessed-top design is the best of the non-snapper models; while, on Tac 1+ and, even more so, Computer Command the response is somewhat mushy. PRO 6000's action is adequate but the niblet-size switch employed is too small for comfort.

Two sticks, Mach III and Premium III, also place "P0" (non-SA) on the case. This lets you rest your main 'trigger finger' during marathon rapid-fire action; and, in applications where handle positioning is critical, means you don't have to worry about jiggling the handle with a button press. Unfortunately for lefties, button placement on the sticks offering this feature favors right-handers only.

All of the standard format units place "P1", the 'aux' or 'second' button, somewhere on the case. All are SA type, except on Premium III and Mach III. (Flight Stick locates an SA-type "P1" at the thumb position on its handle.) Duplicate, left-right "P1" buttons on Tac 1+, PRO 6000, and Computer Command avoid any handedness bias— worth considering if you are left-handed or the stick is intended for school or other multi-user placement. Apparently, no one offers bilateral "P1"s and a front-mounted fire button.

"Centering pull" is a tendency, present in every joystick, to return to center with X,Y values 'pulled' in the direction of most recent handle movement. It was sampled by reading X or Y, moving the handle to extreme right or left (up or down for Y), then easing it back and taking another reading. Horizontal and vertical "Centering Error" numbers presented for each stick reflect average 'pull' over several samplings expressed as percent of half total range. For instance, 10% error corresponds to an average 'pull' of 12.8 in X or Y output. Since there are few applications where centering error below 10% is even noticeable, all six sticks 'passed'. Computer Command and Flight Stick, with less than 2% combined error, rate as virtually "perfect" on this measure.

Given a basically smooth mechanism, probably nothing impacts joystick 'feel' more than centering tension. Light-to-medium tension units, such as Computer Command, Mach III, etc., offer a kind of grease pencil feel, require less gripping force on the handle, and can be operated one-handed, all good attributes for 'general purpose' use, especially if "users"

includes younger children. Experienced action gamers, of course, tend to prefer a heavier tension— something along the lines of the Premium III and Flight Stick. Aside from better centering speed, the player finds it easier to 'feel' stick position; and, with something to 'work against', avoids shadow-boxing fatigue. A good game stick pushes back.

"Effective Centering Tension" is the force, applied at the point where the handle is normally gripped, required to push the handle off center. (Incidentally, tension proved remarkably constant over the full range of handle travel.) The numbers listed provide a basis for comparing tension as well as tendency for 'centering thump'. Higher tension sticks, such as the Premium III, exhibit a more noticeable 'falling into place' feeling, or "thump" when moved through center.

All of the joysticks tested include slides, thumbwheels, or knobs to adjust center X,Y values. This allows the user to compensate for such factors as gradual wear, special centering offsets in some applications, and to correct for shifts when playing some 'normal speed' games with an accelerator or at a 'fast' IIGs setting. Except for Premium III, which was a few scanlines short of Y-axis center on a speeded-up utility, the sticks all displayed plenty of adjustment range.

Possibly, the least reported of all joystick characteristics is something I call "useful range"— the stick swing, in degrees, which actually accounts for a change in X or Y values. The higher this value the better the resolution (less change in X or Y per degree of handle movement) and the easier it will be to target enemy spaceships, touch-up a graphic, etc.. Computer Command and Tac 1+ exhibited smallest useful range; while Mach III and Flight Stick scored best on this measure. Both "Movement Range" and "Useful Range" (sampled for horizontal only) are shown for each joystick.

To the above you can add such considerations as other-machine compatibility, connectors supplied, and ease of case access (for applying occasional squirts of control cleaner). If you plan to do a lot of freehand drawing, presence of a 'centering defeat' option may be of interest.

So, which is the stick for you? Well, as you have, no doubt, been told every time this question comes up about any piece of computer equipment: much depends upon your application. Certainly, individual II+, IIe, IIc owners have numerous factors to consider; whereas educational purchasers for these systems will probably put a high premium on non-handedness. For IIGs types, virtually the only applications area remaining to joysticks is gaming. So, the choice is simple: go for the best game stick you can afford.



## Computer Commander

(Joystick for Apple and IBM, approx. \$30.00 from WICO)

**Size:** 3.87"L x 3.87"W x 1.75"H x 3.25"handle  
**Effective Centering Tension:** approx. 20gm.  
**Handedness:** none  
**Centering Error:** 0.6%H, 0.9%V  
**Centering Defeat:** four flippers on bottom  
**Movement Range:** 50 degrees  
**Useful Range:** 25 degrees  
**Centering Adjust:** slides  
**Connector:** Apple DIN & 16-pin, IBM adapter  
**Case Access:** four screws

This latest in the Computer Command series offers utilitarian understatement in off-white and gray. A mushy in-handle firing button and relatively low resolution result in a 'just fair' game rating; but ambidexterity and very low centering error (the best of any unit tested) are notable plusses. Combined with a low-tension, silky feel these rate an overall "good" for educational, graphics, and other non-game applications.

## Flight Stick

(Joystick for Apple, list price \$74.95, from CH Products)

**Size:** 5.87"L x 6.50"W x 1.75"H x 6.00"handle  
**Effective Centering Tension:** approx. 180gm.  
**Handedness:** none  
**Centering Error:** 1.4%H, 0.5%V  
**Centering Defeat:** none  
**Movement Range:** 55 degrees  
**Useful Range:** 50 degrees  
**Centering Adjust:** thumbwheels  
**Connector:** Apple DIN  
**Case Access:** four screws, remove feet

Featuring black pistol grip, red snap-action switches, and a large-footprint IIGs-colored case, Flight Stick is aimed squarely at action gamers. Boasting very low centering error and excellent resolution, the stick employs an oversized gimble mechanism to achieve a 'starship command' feel exhibiting exceptional smoothness and good speed. Far and away the best *Elite* test performer, Flight Stick rates a surprising "graphics: decent" and a very unsurprising "games: excellent".

## Mach III

(Joystick for Apple, list price \$54.95, from CH Products)

**Size:** 3.75"L x 3.75"W x 1.62"H x 2.50"handle  
**Effective Centering Tension:** approx. 50gm.  
**Handedness:** right  
**Centering Error:** 5.0%H, 3.8%V  
**Centering Defeat:** two slides on top

**Movement Range:** 66 degrees  
**Useful Range:** 50 degrees  
**Centering Adjust:** small knobs  
**Connectors:** Apple DIN  
**Case Access:** four screws, remove feet

Since introducing the fire-button-in-the-handle feature some years back, Mach III has remained the Apple/IBM joystick standard. Excellent resolution, low centering error, and a smooth, medium-tension feel are, clearly, only part of the story. Users appreciate the stick's large, easily gripped handle, classic orange-and-gray on off-white decor, and large, firm buttons. Thus, despite absence of snap-action switches, Mach III rates "good" for gaming and "very good" for graphics and similar applications.

## Premium III

(Joystick for Apple, list price \$44.95, from Kraft Systems)

**Size:** 4.12"L x 4.12"W x 2.25"av.H x 2.25"handle  
**Effective Centering Tension:** approx. 100gm.  
**Handedness:** right  
**Centering Error:** 1.2%H, 3.6%V  
**Centering Defeat:** two flippers on bottom  
**Movement Range:** 60 degrees  
**Useful Range:** 40 degrees  
**Centering Adjust:** thumbwheels  
**Connector:** Apple DIN & 16-pin  
**Case Access:** two screws

With flat-black handle, off-white case, and front-mounted 'aux' button, the Premium III presents a blatantly 'hot' look to match its higher tension, lotus-shaped handle, and snap-action firing switch. Kraft's unique cantilever mechanism imparts a solid, no-flop feel, but also enough right-angle bias to make the stick unsuitable for detailed graphics work. Supplied with a super-supple cable and designed to fit your hand like a glove, Premium III rates a solid "very good" as one of the top game sticks available.

## PRO 6000

(Joystick for Apple and IBM, list price \$29.95, from Control Marketing)

**Size:** 4.75"L x 4.00"W x 1.87"H x 2.50"handle  
**Effective Centering Tension:** approx. 45gm.  
**Handedness:** none  
**Centering Error:** 1.3%H, 2.2%V  
**Centering Defeat:** none  
**Movement Range:** 60 degrees  
**Useful Range:** 40 degrees  
**Centering Adjust:** slides  
**Connector:** Apple DIN & 16-pin, IBM adapter  
**Case Access:** three screws

Trading an on-case firing button for ambidexterity, PRO 6000 amounts to a

credible copy of the popular Mach III done in off-white. Featuring an over-sized, easily gripped handle, this is a smooth-feeling, light-medium tension stick with good resolution and very low centering error. The small, non-snap-action on-handle firing button is a negative; and some graphics users may wish for a centering defeat option; but, on the whole, this stick rates "decent" for gaming and "good-to-very-good" for other applications.

## TAC 1+

(Joystick for Apple and IBM, approx. \$40.00 from Suncom)

**Size:** 4.75"L x 4.25"W x 1.75"H x 3.00"handle  
**Effective Centering Tension:** approx. 25gm.  
**Handedness:** none  
**Centering Error:** 0.6%H, 8.6%V (max observed: 9.5%)  
**Centering Defeat:** none  
**Movement Range:** 50 degrees  
**Useful Range:** 26 degrees  
**Centering Adjust:** slides  
**Connector:** Apple DIN & 16-pin, IBM adapter  
**Case Access:** four screws, remove feet

In Tac 1+, you have adequate performance on such criteria as resolution and centering, and notably elegant lines— even the two 'aux' buttons look like part of the nicely-rounded off-white case. You also, unfortunately, have a square handle! (To appreciate just what this means, try using any joystick without allowing the handle to turn in your grip.) Basically, this is a pretty stick which no one can use for anything but decoration.

In closing, here are three "don't"s to keep in mind.

1: Don't settle for a stick you can't stand the looks of.

2: Don't purchase a stick without a tryout— preferably, of the actual unit you intend to buy. You may need to try two or three of a selected model to find one with just the right feel.

And, of course, whatever you do,

3: Never, Never, let 'em see you sweat!

## Vendors

**CH PRODUCTS** 1225 Stone Drive, San Marcos, CA 92069 (800-624-5804, in CA call 800-262-2004)

**CONTROL MARKETING, INC.** 1461 F Lunt Avenue, Elk Grove, IL 60007 (312-593-6130)

**KRAFT SYSTEMS** 450 W. California Avenue, Vista, CA 92083 (619-724-7146)

**OMNITREND** P.O. Box 3, West Simsbury, CT 06092 (203-658-6917)

**SUNCOM** 260 Holbrook Drive, Wheeling, IL 60090 (312-459-8000)

**WICO** 6400 W. Gross Point Road, Niles, IL 60648 (312-647-7500)





Marc Batchelor

## Fix up your car with the AutoDuel Car Editor

In COMPUTIST #42, Tim Scott presented a character editor for *AutoDuel*, a game to which I'm addicted. Only months previous, I had written a Car editor for the game. However, I did not think that anyone would have much use for the program (except me of course). That is, until I saw Tim's article. At that point, I decided to re-write the program, incorporate some semblance of user-friendliness, and send it to COMPUTIST. This is the product of a couple of days worth of re-write, and some agonizing revelations about the complexity of the *AutoDuel Car*.

The program is pretty much self-explanatory, and menu driven. However, let me explain a few routines. Line 540 (converting HEX - DEC) is only good for 2 digit hex numbers that contain only numbers. (i.e. 1-9, 10-19, 20-29, etc.). Line 570 (converting DEC - HEX) has a different set of restrictions. The result is only designed to take into account 2 digit HEX numbers (i.e. only up to FF). The reason for not making them into full blown routines is simple. It is not necessary, and conservation of code and space is essential.

The *AutoDuel* car information is stored on track \$00, sector \$02. This is where the binary file IOB points to for reading and writing. I didn't incorporate the IOB into poke statements for speed purposes only. Pronto-DOS loads a two sector Binary file MUCH faster than poke statements can poke the IOB in.

Although I am providing this fully commented with remarks, I recommend using a utility program to strip all REMs AFTER you run checksoft on the program. The result is a much faster version. Another good utility is a program that will combine multiple lines into one line when possible. Both of these are available as public domain utilities. I prefer *D-CODE* from Beagle Bros. Micro Software. If you do combine lines and remove remarks, you will notice a tremendous decrease in execution time. It is probably possible (with some modifications I'm sure) to compile this using a basic compiler such as the *Beagle Compiler* or *TASC*. However, since I have *Transwarp*, I have all the speed I'll ever need.

If you have any problems or find any bugs (God forbid), please write a letter to me c/o COMPUTIST and I'll fix it as soon as I can. Enter the binary code and save it as "IOB" then enter and save the BASIC program.

### IOB binary code

```
0300: 01 60 01 00 00 02 11 03      $0392
0308: 00 60 00 00 01 00 00 60      $9BB7
0310: 01 00 01 EF D8 A0 11 A9      $CD25
0318: 03 20 E3 03 A0 00 A9 03      $2D96
0320: 20 D9 03 60 00 00              $111D
```

### Autoduel Car Editor

```
10 D$ = CHR$(13) + CHR$(4):BA = 24576: DIM
   NAS(17)
20 POKE 216,0
30 ONERR GOTO 20
40 POKE 40286,35: POKE 40287,216
50 HOME: A$ = "AUTODUEL^CAR^EDITOR": GOSUB 790:
   VTAB 1: HTAB H: INVERSE: PRINT A$: NORMAL
   : POKE 34,1: A$ = "WRITTEN^ BY^ MARC^
   BATCHELOR": GOSUB 790: VTAB 3: HTAB H:
   PRINT A$
70 FOR T = 1 TO 3000: NEXT
80 IF PEEK(768) <> 1 THEN PRINT D$: "BLOAD^IOB"
90 HOME: VTAB 10: A$ = "PLEASE^PLACE^SIDE^ B^A
   OF^YOUR^": GOSUB 790: HTAB H: PRINT A$
100 VTAB 11: A$ = "AUTODUEL^DISK^IN^DRIVE^1^AND^
   ^": GOSUB 790: HTAB H: PRINT A$
110 A$ = "PRESS^RETURN^TO^LOAD^CAR^": GOSUB 790:
   HTAB H: PRINT A$: A$ = "OR^CTRL-E^IF^CAR^IS^
   LOADED^<^>": GOSUB 790: HTAB H: PRINT A$
120 VTAB 13: HTAB 34: GET TES: IF TES = CHR$(5)
   THEN 140
130 CALL 789
140 HOME
150 REM MENU
160 HOME: GOSUB 810: VTAB 3: PRINT "(1)^CAR^
   NAME^.....": FOR X = 1 TO 16: PRINT
   NAS(X): NEXT:
170 GOSUB 830: VTAB 4: HTAB 1: PRINT "(2)^BODY^
   TYPE^.....":BOS$
180 GOSUB 880: VTAB 5: PRINT "(3)^
   CHASSIS^.....":CH$
190 GOSUB 930: VTAB 6: PRINT "(4)^
   SUSPENSION^.....":SU$
200 GOSUB 980: VTAB 7: PRINT "(5)^MAX^
   WEIGHT^.....":MW$
210 GOSUB 1040: VTAB 8: PRINT "(6)^WEIGHT^
   LEFT^.....":WL$
220 GOSUB 1090: VTAB 9: PRINT "(7)^MAX^
   SPACE^.....":MS$
230 GOSUB 1120: VTAB 10: PRINT "(8)^SPACE^
   LEFT^.....":SL$
240 GOSUB 1150: VTAB 11: PRINT "(9)^HANDLING^
   CLASS^.....":HCS$
250 GOSUB 1180: VTAB 12: PRINT "(10)^
   ACC^.....":ACS$
260 GOSUB 1210: VTAB 13: PRINT "(11)^CAR^
   COST^.....":CC$
270 GOSUB 1270: VTAB 14: PRINT "(12)^BAT^
   CHARGE^.....":BCS$
280 GOSUB 1300: VTAB 15: PRINT "(13)^PWR^PLANT^
   DPTS^.....":PDS$
290 GOSUB 1330: VTAB 16: PRINT "(14)^FL^TIRE^
   DPTS^.....":FLS$
```

```
300 GOSUB 1360: VTAB 17: PRINT "(15)^FR^TIRE^
   DPTS^.....":FRS$
310 GOSUB 1390: VTAB 18: PRINT "(16)^BL^TIRE^
   DPTS^.....":BLS$
320 GOSUB 1420: VTAB 19: PRINT "(17)^BR^TIRE^
   DPTS^.....":BRS$
330 VTAB 22: HTAB 1: PRINT "CHOOSE^NUMBER, <N>^
   FOR^NEXT^SCREEN^": INPUT "OR^<Q>^TO^QUIT^
   AND^SAVE^=>": INS: GOTO 2260
340 HOME: GOSUB 1450: VTAB 3: PRINT "(18)^FNT^
   ARMOR^DPTS^.....":FAS$
350 GOSUB 1480: VTAB 4: PRINT "(19)^BCK^ARMOR^
   DPTS^.....":BDS$
360 GOSUB 1500: VTAB 5: PRINT "(20)^LFT^ARMOR^
   DPTS^.....":LAS$
370 GOSUB 1520: VTAB 6: PRINT "(21)^RT^ARMOR^
   DPTS^.....":RAS$
380 GOSUB 1550: VTAB 7: PRINT "(22)^UDR^ARMOR^
   DPTS^.....":UAS$
390 DSS = "-": FOR T = 1 TO 38: VTAB 8: HTAB T:
   PRINT DSS: NEXT: HTAB 1: PRINT "WEAPON^#"
   : HTAB 12: PRINT "DPTS^": HTAB 18: PRINT
   "LOC^": HTAB 23: PRINT "SPACE^": HTAB 30:
   PRINT "ACT^": HTAB 35: PRINT "AMMO^"
400 DSS = "=": FOR T = 1 TO 38: VTAB 10: HTAB T:
   PRINT DSS: NEXT: HTAB 1: VTAB 11: PRINT
   "1^": HTAB 1: VTAB 12: PRINT "2^": HTAB 1:
   VTAB 13: PRINT "3^": HTAB 1: VTAB 14: PRINT
   "4^": HTAB 1: VTAB 15: PRINT "5^": HTAB 1:
   VTAB 16: PRINT "6^"
410 HTAB 1: VTAB 17: PRINT "7^": HTAB 1: VTAB 18:
   PRINT "8^": HTAB 1: VTAB 19: PRINT "9^":
   HTAB 1: VTAB 20: PRINT "0^"
420 GOSUB 1560: VTAB 11: HTAB 3: PRINT WES$: HTAB
   12: PRINT DPS$: HTAB 18: PRINT LOS$: HTAB
   23: PRINT SPS$: HTAB 30: PRINT ACS$: HTAB
   35: PRINT AMS$
430 GOSUB 1630: VTAB 12: HTAB 3: PRINT WES$: HTAB
   12: PRINT DPS$: HTAB 18: PRINT LOS$: HTAB
   23: PRINT SPS$: HTAB 30: PRINT ACS$: HTAB
   35: PRINT AMS$
440 GOSUB 1700: VTAB 13: HTAB 3: PRINT WES$: HTAB
   12: PRINT DPS$: HTAB 18: PRINT LOS$: HTAB
   23: PRINT SPS$: HTAB 30: PRINT ACS$: HTAB
   35: PRINT AMS$
450 GOSUB 1770: VTAB 14: HTAB 3: PRINT WES$: HTAB
   12: PRINT DPS$: HTAB 18: PRINT LOS$: HTAB
   23: PRINT SPS$: HTAB 30: PRINT ACS$: HTAB
   35: PRINT AMS$
460 GOSUB 1840: VTAB 15: HTAB 3: PRINT WES$: HTAB
   12: PRINT DPS$: HTAB 18: PRINT LOS$: HTAB
   23: PRINT SPS$: HTAB 30: PRINT ACS$: HTAB
   35: PRINT AMS$
470 GOSUB 1910: VTAB 16: HTAB 3: PRINT WES$: HTAB
   12: PRINT DPS$: HTAB 18: PRINT LOS$: HTAB
   23: PRINT SPS$: HTAB 30: PRINT ACS$: HTAB
   35: PRINT AMS$
480 GOSUB 1980: VTAB 17: HTAB 3: PRINT WES$: HTAB
   12: PRINT DPS$: HTAB 18: PRINT LOS$: HTAB
   23: PRINT SPS$: HTAB 30: PRINT ACS$: HTAB
   35: PRINT AMS$
```



```

490 GOSUB 2050:VTAB 18:HTAB 3:PRINT WE$:HTAB
12:PRINT DP$:HTAB 18:PRINT LOS:HTAB
23:PRINT SP$:HTAB 30:PRINT ACS:HTAB
35:PRINT AMS
500 GOSUB 2120:VTAB 19:HTAB 3:PRINT WE$:HTAB
12:PRINT DP$:HTAB 18:PRINT LOS:HTAB
23:PRINT SP$:HTAB 30:PRINT ACS:HTAB
35:PRINT AMS
510 GOSUB 2190:VTAB 20:HTAB 3:PRINT WE$:HTAB
12:PRINT DP$:HTAB 18:PRINT LOS:HTAB
23:PRINT SP$:HTAB 30:PRINT ACS:HTAB
35:PRINT AMS
520 VTAB 22:HTAB 1:PRINT "CHOOSE^NUMBER.^<P>^
FOR^PREVIOUS^SCREEN":INPUT "OR^<Q>^TO^
QUIT^AND^SAVE=>":IN$:GOTO 3220
530 END
540 REM CONVERT HEX - DEC
550 A = ((T >= 90) + (T >= 80) + (T >= 70) + (T
>= 60) + (T >= 50) + (T >= 40) + (T >= 30)
+ (T >= 20) + (T >= 10))
560 RETURN
570 REM CONVERT DEC - HEX
580 A = INT (VA / 16):B = A * 16:C = VA - B
590 IF C = 10 THEN CS = "A":RETURN
600 IF C = 11 THEN CS = "B":RETURN
610 IF C = 12 THEN CS = "C":RETURN
620 IF C = 13 THEN CS = "D":RETURN
630 IF C = 14 THEN CS = "E":RETURN
640 IF C = 15 THEN CS = "F":RETURN
650 CS = STR$(C):RETURN
660 REM WEAPON STATISTICS
670 IF L1 > 4 THEN L1 = 4
680 IF W1 > 12 THEN W1 = 12
690 WSS = "M^GUN^FLTHR^RL^RR^AT^
^LASER^M^DROP^S^DROP^S^SCREENPAINT^
^OIL^HR^"
700 WES = MID$(WSS,(W1 * 8 + 1),8)
710 LSS = "FNTBCKLFRTR"
720 LOS = MID$(LSS,(L1 * 3 + 1),3)
730 RETURN
740 REM PRINT WEAPON CHOICES
750 VTAB 6:PRINT "1)^WEAPON":PRINT "2)^DPTS"
:PRINT "3)^LOCATION":PRINT "4)^SPACE"
:PRINT "5)^ACTIVE/INACTIVE":PRINT "6)^
AMMO":RETURN
760 REM CALCULATE RELATIVE ADDRESS
770 IF IN = 0 THEN IN = 10
780 A1 = (BA + 176):AD = (8 * (IN - 1)) + A1:
RETURN
790 REM CENTER STRING ROUTINE
800 H = 21 - (LEN (AS) / 2):RETURN
810 REM GET CAR NAME
820 X = 1:FOR NA = BA TO (BA + 16):NAS(X) = CHR$(
PEEK (NA)):X = X + 1:NEXT:RETURN
830 REM GET BODY
840 BO = PEEK (BA + 16)
850 B1$ = "SUBCOMPACT^COMPACT^MID-SIZED^
^LUXURY^STATION^WAGON^PICKUP^
^VAN^"
860 BO$ = MID$(B1$,(BO * 13 + 1),13)
870 RETURN
880 REM GET CHASSIS
890 CH = PEEK (BA + 17)

```

```

900 C1$ = "LIGHT^STANDARD^HEAVY^
EXTRA-HEAVY"
910 CHS = MID$(C1$,(CH * 11 + 1),11)
920 RETURN
930 REM GET SUSPENSION
940 SU = PEEK (BA + 18)
950 S1$ = "LIGHT^IMPROVEDHEAVY^"
960 SU$ = MID$(S1$,(SU * 8 + 1),8)
970 RETURN
980 REM GET MAXIMUM WEIGHT
990 W1 = PEEK (BA + 20):W2 = PEEK (BA + 19)
1000 VA = W1:GOSUB 570:W3$ = STR$(A) + CS
1010 VA = W2:GOSUB 570:W4$ = STR$(A) + CS
1020 MWS = W3$ + W4$
1030 RETURN
1040 REM GET WEIGHT LEFT
1050 L1 = PEEK (BA + 22):L2 = PEEK (BA + 21)
1060 VA = L1:GOSUB 570:L3$ = STR$(A) + CS
1070 VA = L2:GOSUB 570:L4$ = STR$(A) + CS
1080 WL$ = L3$ + L4$:RETURN
1090 REM GET MAX SPACE
1100 MS = PEEK (BA + 23)
1110 VA = MS:GOSUB 570:MSS = STR$(A) + CS:
RETURN
1120 REM GET SPACE LEFT
1130 SL = PEEK (BA + 24)
1140 VA = SL:GOSUB 570:SL$ = STR$(A) + CS:
RETURN
1150 REM GET HANDLING CLASS
1160 HC = PEEK (BA + 25)
1170 VA = HC:GOSUB 570:HC$ = STR$(A) + CS:
RETURN
1180 REM GET ACC
1190 AC = PEEK (BA + 27)
1200 VA = AC:GOSUB 570:AC$ = STR$(A) + CS:
RETURN
1210 REM GET CAR COST
1220 C1 = PEEK (BA + 31):C2 = PEEK (BA + 30):C3
= PEEK (BA + 29)
1230 VA = C1:GOSUB 570:C1$ = STR$(A) + CS
1240 VA = C2:GOSUB 570:C2$ = STR$(A) + CS
1250 VA = C3:GOSUB 570:C3$ = STR$(A) + CS
1260 CC$ = C1$ + C2$ + C3$:RETURN
1270 REM GET BATTERY CHARGE
1280 BC = PEEK (BA + 35)
1290 VA = BC:GOSUB 570:BC$ = STR$(A) + CS:
RETURN
1300 REM GET PWR PLANT DPTS
1310 PD = PEEK (BA + 97)
1320 VA = PD:GOSUB 570:PD$ = STR$(A) + CS:
RETURN
1330 REM GET FNT LT TIRE DPTS
1340 FL = PEEK (BA + 105)
1350 VA = FL:GOSUB 570:FL$ = STR$(A) + CS:
RETURN
1360 REM GET FNT RT TIRE DPTS
1370 FR = PEEK (BA + 113)
1380 VA = FR:GOSUB 570:FR$ = STR$(A) + CS:
RETURN
1390 REM GET BCK LT TIRE DPTS
1400 BL = PEEK (BA + 121)
1410 VA = BL:GOSUB 570:BL$ = STR$(A) + CS:
RETURN
1420 REM GET BCK RT TIRE DPTS

```

```

1430 BR = PEEK (BA + 129)
1440 VA = BR:GOSUB 570:BR$ = STR$(A) + CS:
RETURN
1450 REM GET FNT ARMOR DPTS
1460 FA = PEEK (BA + 137)
1470 VA = FA:GOSUB 570:FA$ = STR$(A) + CS:
RETURN
1480 REM GET BCK ARMOR DPTS
1490 BD = PEEK (BA + 145):VA = BD:GOSUB 570:BD$
= STR$(A) + CS:RETURN
1500 REM GET LFT ARMOR DPTS
1510 LA = PEEK (BA + 153):VA = LA:GOSUB 570:LAS
= STR$(A) + CS:RETURN
1520 REM GET RT ARMOR DPTS
1530 RA = PEEK (BA + 161):VA = RA:GOSUB 570:RAS
= STR$(A) + CS:RETURN
1540 REM GET UDR ARMOR DPTS
1550 UA = PEEK (BA + 169):VA = UA:GOSUB 570:UAS
= STR$(A) + CS:RETURN
1560 REM GET WEAPON #1
1570 W1 = PEEK (BA + 176):L1 = PEEK (BA + 179):
GOSUB 660:DP = PEEK (BA + 177):SP = PEEK (BA
+ 180):AC = PEEK (BA + 181):AM = PEEK (BA +
182)
1580 VA = DP:GOSUB 570:DPS = STR$(A) + CS:VA =
SP:GOSUB 570:SPS = STR$(A) + CS:VA = AC:
GOSUB 570:ACS = STR$(A) + CS:VA = AM:GOSUB
570:AMS = STR$(A) + CS
1590 IF ACS <> "80" THEN ACS = "NO":GOTO 1610
1600 ACS = "YES"
1610 IF AMS = "13D" OR AMS = "15F" THEN AMS = ""
:RETURN
1620 RETURN
1630 REM GET WEAPON #2
1640 W1 = PEEK (BA + 184):L1 = PEEK (BA + 187):
GOSUB 660:DP = PEEK (BA + 185):SP = PEEK (BA
+ 188):AC = PEEK (BA + 189):AM = PEEK (BA +
190)
1650 VA = DP:GOSUB 570:DPS = STR$(A) + CS:VA =
SP:GOSUB 570:SPS = STR$(A) + CS:VA = AC:
GOSUB 570:ACS = STR$(A) + CS:VA = AM:GOSUB
570:AMS = STR$(A) + CS
1660 IF ACS <> "80" THEN ACS = "NO":GOTO 1680
1670 ACS = "YES"
1680 IF AMS = "13D" OR AMS = "15F" THEN AMS = ""
:RETURN
1690 RETURN
1700 REM GET WEAPON #3
1710 W1 = PEEK (BA + 192):L1 = PEEK (BA + 195):
GOSUB 660:DP = PEEK (BA + 193):SP = PEEK (BA
+ 196):AC = PEEK (BA + 197):AM = PEEK (BA +
198)
1720 VA = DP:GOSUB 570:DPS = STR$(A) + CS:VA =
SP:GOSUB 570:SPS = STR$(A) + CS:VA = AC:
GOSUB 570:ACS = STR$(A) + CS:VA = AM:GOSUB
570:AMS = STR$(A) + CS
1730 IF ACS <> "80" THEN ACS = "NO":GOTO 1750
1740 ACS = "YES"
1750 IF AMS = "13D" OR AMS = "15F" THEN AMS = ""
:RETURN
1760 RETURN
1770 REM GET WEAPON #4

```





# Readers Data Exchange



```

1780 W1 = PEEK (BA + 200) : L1 = PEEK (BA + 203) :
GOSUB 660 : DP = PEEK (BA + 201) : SP = PEEK (BA
+ 204) : AC = PEEK (BA + 205) : AM = PEEK (BA +
206)
1790 VA = DP : GOSUB 570 : DPS = STR$ (A) + C$ : VA =
SP : GOSUB 570 : SP$ = STR$ (A) + C$ : VA = AC :
GOSUB 570 : ACS = STR$ (A) + C$ : VA = AM : GOSUB
570 : AMS = STR$ (A) + C$
1800 IF ACS <> "80" THEN ACS = "NO" : GOTO 1820
1810 ACS = "YES"
1820 IF AMS = "13D" OR AMS = "15F" THEN AMS = "" :
: RETURN
1830 RETURN
1840 REM GET WEAPON #5
1850 W1 = PEEK (BA + 208) : L1 = PEEK (BA + 211) :
GOSUB 660 : DP = PEEK (BA + 209) : SP = PEEK (BA
+ 212) : AC = PEEK (BA + 213) : AM = PEEK (BA +
214)
1860 VA = DP : GOSUB 570 : DPS = STR$ (A) + C$ : VA =
SP : GOSUB 570 : SP$ = STR$ (A) + C$ : VA = AC :
GOSUB 570 : ACS = STR$ (A) + C$ : VA = AM : GOSUB
570 : AMS = STR$ (A) + C$
1870 IF ACS <> "80" THEN ACS = "NO" : GOTO 1890
1880 ACS = "YES"
1890 IF AMS = "13D" OR AMS = "15F" THEN AMS = "" :
: RETURN
1900 RETURN
1910 REM GET WEAPON #6
1920 W1 = PEEK (BA + 216) : L1 = PEEK (BA + 219) :
GOSUB 660 : DP = PEEK (BA + 217) : SP = PEEK (BA
+ 220) : AC = PEEK (BA + 221) : AM = PEEK (BA +
222)
1930 VA = DP : GOSUB 570 : DPS = STR$ (A) + C$ : VA =
SP : GOSUB 570 : SP$ = STR$ (A) + C$ : VA = AC :
GOSUB 570 : ACS = STR$ (A) + C$ : VA = AM : GOSUB
570 : AMS = STR$ (A) + C$
1940 IF ACS <> "80" THEN ACS = "NO" : GOTO 1960
1950 ACS = "YES"
1960 IF AMS = "13D" OR AMS = "15F" THEN AMS = "" :
: RETURN
1970 RETURN
1980 REM GET WEAPON #7
1990 W1 = PEEK (BA + 224) : L1 = PEEK (BA + 227) :
GOSUB 660 : DP = PEEK (BA + 225) : SP = PEEK (BA
+ 228) : AC = PEEK (BA + 229) : AM = PEEK (BA +
230)
2000 VA = DP : GOSUB 570 : DPS = STR$ (A) + C$ : VA =
SP : GOSUB 570 : SP$ = STR$ (A) + C$ : VA = AC :
GOSUB 570 : ACS = STR$ (A) + C$ : VA = AM : GOSUB
570 : AMS = STR$ (A) + C$
2010 IF ACS <> "80" THEN ACS = "NO" : GOTO 2030
2020 ACS = "YES"
2030 IF AMS = "13D" OR AMS = "15F" THEN AMS = "" :
: RETURN
2040 RETURN
2050 REM GET WEAPON #8
2060 W1 = PEEK (BA + 232) : L1 = PEEK (BA + 235) :
GOSUB 660 : DP = PEEK (BA + 233) : SP = PEEK (BA
+ 236) : AC = PEEK (BA + 237) : AM = PEEK (BA +
238)
2070 VA = DP : GOSUB 570 : DPS = STR$ (A) + C$ : VA =
SP : GOSUB 570 : SP$ = STR$ (A) + C$ : VA = AC :
GOSUB 570 : ACS = STR$ (A) + C$ : VA = AM : GOSUB
570 : AMS = STR$ (A) + C$
2080 IF ACS <> "80" THEN ACS = "NO" : GOTO 2100
2090 ACS = "YES"
2100 IF AMS = "13D" OR AMS = "15F" THEN AMS = "" :
: RETURN
2110 RETURN
2120 REM GET WEAPON #9
2130 W1 = PEEK (BA + 240) : L1 = PEEK (BA + 243) :
GOSUB 660 : DP = PEEK (BA + 241) : SP = PEEK (BA
+ 244) : AC = PEEK (BA + 245) : AM = PEEK (BA +
246)
2140 VA = DP : GOSUB 570 : DPS = STR$ (A) + C$ : VA =
SP : GOSUB 570 : SP$ = STR$ (A) + C$ : VA = AC :
GOSUB 570 : ACS = STR$ (A) + C$ : VA = AM : GOSUB
570 : AMS = STR$ (A) + C$
2150 IF ACS <> "80" THEN ACS = "NO" : GOTO 2170
2160 ACS = "YES"
2170 IF AMS = "13D" OR AMS = "15F" THEN AMS = "" :
: RETURN
2180 RETURN
2190 REM GET WEAPON #0
2200 W1 = PEEK (BA + 248) : L1 = PEEK (BA + 251) :
GOSUB 660 : DP = PEEK (BA + 249) : SP = PEEK (BA
+ 252) : AC = PEEK (BA + 253) : AM = PEEK (BA +
254)
2210 VA = DP : GOSUB 570 : DPS = STR$ (A) + C$ : VA =
SP : GOSUB 570 : SP$ = STR$ (A) + C$ : VA = AC :
GOSUB 570 : ACS = STR$ (A) + C$ : VA = AM : GOSUB
570 : AMS = STR$ (A) + C$
2220 IF ACS <> "80" THEN ACS = "NO" : GOTO 2240
2230 ACS = "YES"
2240 IF AMS = "13D" OR AMS = "15F" THEN AMS = "" :
: RETURN
2250 RETURN
2260 REM *****FIRST SCREEN CHOICES*****
2270 IF IN$ = "N" THEN GOTO 340
2280 IF IN$ = "" THEN GOTO 330
2290 IF IN$ = "Q" THEN GOTO 3690
2300 IF VAL (IN$) > 17 OR VAL (IN$) < 1 GOTO 330
2310 IN = VAL (IN$) : IF IN > 1 GOTO 2380
2320 REM CHANGE CAR NAME
2330 HOME : VTAB 4 : PRINT "NEW*NAME*(16*CHARS*
MAX)" : INPUT "" : NBS
2340 IF NBS = "" GOTO 150
2350 IF LEN (NBS) > 16 THEN 2310
2360 IF LEN (NBS) < 16 THEN SP$ = "" : TE = 16 -
LEN (NBS) : FOR T = 1 TO TE : NBS = NBS + SP$ :
NEXT
2370 FOR T = 1 TO 16 : NA$(T) = MID$ (NBS, T, 1) :
NEXT : FOR T = 1 TO 16 : POKE ((BA - 1) + T) :
ASC (NA$(T)) + 128 : NEXT : GOTO 150
2380 IF IN > 2 GOTO 2440
2390 REM CHANGE BODY TYPE
2400 HOME : VTAB 4 : PRINT "NEW*BODY*TYPE*=>" :
PRINT "1)*SUBCOMPACT" : PRINT "2)*
COMPACT" : PRINT "3)*MID-SIZED" : PRINT
"4)*LUXURY" : PRINT "5)*STAT ION*WAGON" :
PRINT "6)*PICKUP" : PRINT "7)*VAN"
2410 VTAB 4 : HTAB 17 : INPUT "" : BOS : IF BOS = ""
GOTO 150
2420 IF BOS <> "1" AND BOS <> "2" AND BOS <> "3"
AND BOS <> "4" AND BOS <> "5" AND BOS <> "6"
AND BOS <> "7" GOTO 2400
2430 BO = VAL (BOS) : POKE (BA + 16) : (BO - 1) : HOME
: GOTO 150
2440 IF IN > 3 GOTO 2500
2450 REM CHANGE CHASSIS
2460 HOME : VTAB 4 : PRINT "NEW*CHASSIS*=>" :
PRINT "1)*LIGHT" : PRINT "2)*STANDARD" :
PRINT "3)*HEAVY" : PRINT "4)*EXTRA-HEAVY"
2470 VTAB 4 : HTAB 15 : INPUT "" : CH$ : IF CH$ = ""
GOTO 150
2480 IF CH$ <> "1" AND CH$ <> "2" AND CH$ <> "3"
AND CH$ <> "4" GOTO 2460
2490 CH = VAL (CH$) : POKE (BA + 17) : (CH - 1) : HOME
: GOTO 150
2500 IF IN > 4 GOTO 2550
2510 REM CHANGE SUSPENSION
2520 HOME : VTAB 4 : PRINT "NEW*SUSPENSION*=>"
: PRINT "1)*LIGHT" : PRINT "2)*IMPROVED"
: PRINT "3)*HEAVY" : VTAB 4 : HTAB 18 : INPUT
"" : SUS : IF SUS = "" GOTO 150
2530 IF SUS <> "1" AND SUS <> "2" AND SUS <> "3"
GOTO 2520
2540 SU = VAL (SUS) : POKE (BA + 18) : (SU - 1) : HOME
: GOTO 150
2550 IF IN > 5 GOTO 2620
2560 REM CHANGE MAXIMUM WEIGHT
2570 HOME : VTAB 4 : INPUT "NEW*MAX*WEIGHT*(9999*
MAX)*=>" : MWS
2580 IF MWS = "" GOTO 150
2590 IF LEN (MWS) > 4 GOTO 2570
2600 W1$ = LEFT$ (MWS, 2) : W2$ = RIGHT$
(MWS, 2) : W1 = VAL (W1$) : W2 = VAL (W2$) : T =
W1 : GOSUB 540 : W1 = T + (6 * A) : T = W2 : GOSUB
540 : W2 = T + (6 * A)
2610 POKE (BA + 20) : W1 : POKE (BA + 19) : W2 : HOME
: GOTO 150
2620 IF IN > 6 GOTO 2680
2630 REM CHANGE WEIGHT LEFT
2640 HOME : VTAB 4 : INPUT "NEW*WEIGHT*LEFT*(9999*
MAX)*=>" : WLS : IF WLS = "" GOTO 150
2650 IF LEN (WLS) > 4 GOTO 2640
2660 L1$ = LEFT$ (WLS, 2) : L2$ = RIGHT$
(WLS, 2) : L1 = VAL (L1$) : L2 = VAL (L2$) : T =
L1 : GOSUB 540 : L1 = T + (6 * A) : T = L2 : GOSUB
540 : L2 = T + (6 * A) : POKE (BA + 22) : L1 :
POKE (BA + 21) : L2
2670 HOME : GOTO 150
2680 IF IN > 7 GOTO 2730
2690 REM CHANGE MAX SPACE
2700 HOME : VTAB 4 : INPUT "NEW*MAX*SPACE*(99*
MAX)*=>" : MSS : IF MSS = "" GOTO 150
2710 IF LEN (MSS) > 2 GOTO 2700
2720 MS = VAL (MSS) : T = MS : GOSUB 540 : MS = T + (6
* A) : POKE (BA + 23) : MS : HOME : GOTO 150
2730 IF IN > 8 GOTO 2780
2740 REM CHANGE SPACE LEFT
2750 HOME : VTAB 4 : INPUT "NEW*SPACE*LEFT*(99*
MAX)*=>" : SL$ : IF SL$ = "" GOTO 150
2760 IF LEN (SL$) > 4 GOTO 2750
2770 SL = VAL (SL$) : T = SL : GOSUB 540 : SL = T + (6
* A) : POKE (BA + 24) : SL : HOME : GOTO 150
2780 IF IN > 9 GOTO 2830
2790 REM CHANGE HANDLING CLASS
2800 HOME : VTAB 4 : INPUT "NEW*H.*CLASS*(0-3)*
=>" : HCS : IF HCS = "" GOTO 150
2810 IF HCS <> "0" AND HCS <> "1" AND HCS <> "2"
AND HCS <> "3" GOTO 2800

```



# Readers Data EXchange



```

2820 HC = VAL (HC$) : POKE (BA + 25) . HC : HOME :
      GOTO 150
2830 IF IN > 10 GOTO 2880
2840 REM CHANGE ACCELERATION
2850 HOME : VTAB 4 : INPUT "NEW ACC (0-2) =>"
      : ACS : IF ACS = "" GOTO 150
2860 IF ACS <> "0" AND ACS <> "1" AND ACS <> "2"
      GOTO 2850
2870 AC = VAL (ACS) : T = AC : GOSUB 540 : AC = T + (6
      * A) : POKE (BA + 27) . AC : HOME : GOTO 150
2880 IF IN > 11 GOTO 2940
2890 REM CHANGE CAR COST
2900 HOME : VTAB 4 : INPUT "NEW CAR COST (999999*
      MAX) =>" : CCS : IF CCS = "" GOTO 150
2910 IF LEN (CC$) > 6 GOTO 2900
2920 CAS = LEFT$ (CC$, 4) : C1$ = LEFT$
      (CAS, 2) : C2$ = RIGHT$ (CAS, 2) : C3$ = RIGHT$
      (CC$, 2) : C1 = VAL (C1$) : C2 = VAL (C2$) : C3 =
      VAL (C3$)
2930 T = C1 : GOSUB 540 : C1 = T + (6 * A) : POKE (BA
      + 31) . C1 : T = C2 : GOSUB 540 : C2 = T + (6 * A) :
      POKE (BA + 30) . C2 : T = C3 : GOSUB 540 : C3 = T
      + (6 * A) : POKE (BA + 29) . C3 : HOME : GOTO
      150
2940 IF IN > 12 GOTO 2990
2950 REM CHANGE BATTERY CHARGE
2960 HOME : VTAB 4 : INPUT "NEW BAT CHARGE (99
      MAX) =>" : BC$ : IF BC$ = "" GOTO 150
2970 IF LEN (BC$) > 2 GOTO 2960
2980 BC = VAL (BC$) : T = BC : GOSUB 540 : BC = T + (6
      * A) : POKE (BA + 35) . BC : HOME : GOTO 150
2990 IF IN > 13 GOTO 3040
3000 REM CHANGE PWR PLANT DAMAGE PTS
3010 HOME : VTAB 4 : INPUT "NEW PWR PLANT DPTS (
      99*MAX) =>" : PD$ : IF PD$ = "" GOTO 150
3020 IF LEN (PD$) > 2 GOTO 3010
3030 PD = VAL (PD$) : T = PD : GOSUB 540 : PD = T + (6
      * A) : POKE (BA + 97) . PD : POKE (BA + 98) . PD :
      HOME : GOTO 150
3040 IF IN > 14 GOTO 3090
3050 REM CHANGE FL TIRE DPTS
3060 HOME : VTAB 4 : INPUT "NEW FL TIRE DPTS (99
      MAX) =>" : FL$ : IF FL$ = "" GOTO 150
3070 IF LEN (FL$) > 2 GOTO 3060
3080 FL = VAL (FL$) : T = FL : GOSUB 540 : FL = T + (6
      * A) : POKE (BA + 105) . FL : POKE (BA + 106) . FL :
      HOME : GOTO 150
3090 IF IN > 15 GOTO 3140
3100 REM CHANGE FR TIRE DPTS
3110 HOME : VTAB 4 : INPUT "NEW FR TIRE DPTS (99
      MAX) =>" : FR$ : IF FR$ = "" GOTO 150
3120 IF LEN (FR$) > 2 GOTO 3110
3130 FR = VAL (FR$) : T = FR : GOSUB 540 : FR = T + (6
      * A) : POKE (BA + 113) . FR : POKE (BA + 114) . FR :
      HOME : GOTO 150
3140 IF IN > 16 GOTO 3190
3150 REM CHANGE BL TIRE DPTS
3160 HOME : VTAB 4 : INPUT "NEW BL TIRE DPTS (99
      MAX) =>" : BL$ : IF BL$ = "" GOTO 150
3170 IF LEN (BL$) > 2 GOTO 3160
3180 BL = VAL (BL$) : T = BL : GOSUB 540 : BL = T + (6
      * A) : POKE (BA + 121) . BL : POKE (BA + 122) . BL :
      HOME : GOTO 150
3185 REM CHANGE BR TIRE DPTS

```

```

3190 HOME : VTAB 4 : INPUT "NEW BR TIRE DPTS (99
      MAX) =>" : BR$ : IF BR$ = "" GOTO 150
3200 IF LEN (BR$) > 2 GOTO 3190
3210 BR = VAL (BR$) : T = BR : GOSUB 540 : BR = T + (6
      * A) : POKE (BA + 129) . BR : POKE (BA + 130) . BR :
      HOME : GOTO 150
3220 REM *** SECOND SCREEN CHOICES ***
3230 IF IN$ = "P" THEN GOTO 150
3240 IF IN$ = "" GOTO 520
3250 IF IN$ = "Q" THEN GOTO 3690
3260 FOR T = 0 TO 9 : IF VAL (IN$) <> T THEN NEXT
      : GOTO 3280
3270 GOTO 3290
3280 FOR T = 18 TO 22 : IF VAL (IN$) <> T THEN NEXT
      : GOTO 520
3290 IN = VAL (IN$) : IF IN > 9 GOTO 3500
3300 HOME : VTAB 4 : PRINT "WEAPON#:" : IN : GOSUB
      740 : GOSUB 760 : PRINT : INPUT "CHOICE#"
      : CH$ : IF CH$ = "" GOTO 340
3310 IF VAL (CH$) > 6 OR VAL (CH$) < 1 GOTO 3300
3320 CH = VAL (CH$) : ON CH GOTO
      3330, 3390, 3400, 3440, 3460, 3480
3330 HOME : PRINT : PRINT "0" * "MACHINE GUN" :
      PRINT "1" * "FLAME THROWER" : PRINT "2" *
      "ROCKET LAUNCHER" : PRINT "3" * "RECOILLESS
      RIFLE" : PRINT "4" * "ANTI-TANK GUN" : PRINT
      "5" * "LASER" : PRINT "6" * "MINE DROPPER" :
      PRINT "7" * "SPIKE DROPPER" : PRINT "8" *
      "SMOKE"
3340 PRINT "9" * "PAINT SPRAYER" : PRINT "10" * "OIL
      JET" : PRINT "11" * "HEAVY ROCKET" : PRINT :
      INPUT "CHOICE#" : CH$ : IF CH$ = "" GOTO 340
3350 IF VAL (CH$) > 11 OR VAL (CH$) < 1 GOTO 3370
3360 GOTO 3380
3370 IF CH$ <> "0" GOTO 3330
3380 CH = VAL (CH$) : POKE AD, CH : GOTO 340
3390 HOME : PRINT : INPUT "NEW DPTS (99*MAX) =>"
      : DP$ : T = VAL (DP$) : GOSUB 540 : DP = T + (6 *
      A) : POKE (AD + 1) . DP : POKE (AD + 2) . DP : HOME
      : GOTO 340
3400 HOME : PRINT : PRINT "NEW LOCATION" : PRINT
      : PRINT "0" * "FRONT" : PRINT "1" * "BACK" :
      PRINT "2" * "LEFT" : PRINT "3" * "RIGHT" : PRINT
      : INPUT "CHOICE#" : LOS : IF VAL (LOS) > 3 OR
      VAL (LOS) < 1 GOTO 3420
3410 GOTO 3430
3420 IF LOS <> "0" THEN 3400
3430 T = VAL (LOS) : GOSUB 540 : LO = T + (6 * A) :
      POKE (AD + 3) . LO : HOME : GOTO 340
3440 HOME : PRINT : INPUT "NEW SPACE (99*MAX)
      =>" : SP$ : IF LEN (SP$) > 2 GOTO 3440
3450 T = VAL (SP$) : GOSUB 540 : SP = T + (6 * A) :
      POKE (AD + 4) . SP : HOME : GOTO 340
3460 IF PEEK (AD + 5) = 80 THEN POKE (AD + 5) . 0 :
      HOME : GOTO 340
3470 IF PEEK (AD + 5) <> 80 THEN T = 80 : GOSUB
      540 : T = T + (6 * A) : POKE (AD + 5) . T : HOME
      : GOTO 340
3480 HOME : PRINT : INPUT "NEW AMMO (99*MAX) =>"
      : AM$ : IF LEN (AM$) > 2 GOTO 3480
3490 T = VAL (AM$) : GOSUB 540 : AM = T + (6 * A) :
      POKE (AD + 6) . AM : HOME : GOTO 340
3500 IF IN > 18 GOTO 3550
3510 REM CHANGE FRONT ARMOR

```

```

3520 HOME : PRINT : INPUT "NEW FRONT ARMOR (99
      MAX) =>" : FAS : IF LEN (FAS) > 2 GOTO 3520
3530 T = VAL (FAS) : GOSUB 540 : FA = T + (6 * A) :
      POKE (BA + 137) . FA : POKE (BA + 138) . FA :
      HOME : GOTO 340
3540 REM CHANGE BACK ARMOR
3550 IF IN > 19 GOTO 3590
3560 HOME : PRINT : INPUT "NEW BACK ARMOR (99
      MAX) =>" : BDS : IF LEN (BDS) > 2 GOTO 3550
3570 T = VAL (BDS) : GOSUB 540 : BD = T + (6 * A) :
      POKE (BA + 145) . BD : POKE (BA + 146) . BD :
      HOME : GOTO 340
3580 REM CHANGE LEFT ARMOR
3590 IF IN > 20 GOTO 3620
3600 HOME : PRINT : INPUT "NEW LEFT ARMOR (99
      MAX) =>" : LAS : IF LEN (LAS) > 2 GOTO 3600
3610 T = VAL (LAS) : GOSUB 540 : LA = T + (6 * A) :
      POKE (BA + 153) . LA : POKE (BA + 154) . LA :
      HOME : GOTO 340
3620 REM CHANGE RIGHT ARMOR
3630 IF IN > 21 GOTO 3660
3640 HOME : PRINT : INPUT "NEW RIGHT ARMOR (99
      MAX) =>" : RAS : IF LEN (RAS) > 2 GOTO 3640
3650 T = VAL (RAS) : GOSUB 540 : RA = T + (6 * A) :
      POKE (BA + 161) . RA : POKE (BA + 162) . RA :
      HOME : GOTO 340
3660 REM CHANGE UNDER ARMOR
3670 HOME : PRINT : INPUT "NEW UNDER ARMOR (99
      MAX) =>" : UAS : IF LEN (UAS) > 2 GOTO 3670
3680 T = VAL (UAS) : GOSUB 540 : UA = T + (6 * A) :
      POKE (BA + 169) . UA : POKE (BA + 169) . UA :
      HOME : GOTO 340
3685 REM QUIT AND SAVE ATTRIBUTES
3690 HOME : PRINT : INPUT "SAVE CAR? Y/N"
      : QTS : IF QTS <> "Y" AND QTS <> "N" GOTO
      3690
3700 IF QTS = "N" THEN TEXT : HOME : NEW
3710 POKE 780, 2 : HOME : AS = "PLEASE PLACE
      SIDE B OF YOUR" : GOSUB 790 : VTAB 6 :
      HTAB H : PRINT AS : AS = "AUTODUEL DISK IN
      DRIVE 1 AND" : GOSUB 790 : VTAB 7 : HTAB H :
      PRINT AS
3720 AS = "PRESS RETURN TO CONTINUE." :
      GOSUB 790 : VTAB 8 : HTAB H : PRINT AS : AS =
      "ANY OTHER KEY ABANDONS" : GOSUB 790 :
      VTAB 9 : HTAB H : PRINT AS
3730 VTAB 9 : HTAB (H + 23) : GET SAS : IF SAS <>
      CHR$ (13) THEN GOTO 150
3740 CALL 789 : TEXT : HOME : NEW

```

## Checksums

10	- \$E297	1900	- \$1CF2
20	- \$9261	1910	- \$BD81
30	- \$3DD3	1920	- \$3877
40	- \$7957	1930	- \$BF10
50	- \$798C	1940	- \$DBAF
70	- \$AEE3	1950	- \$38D4
80	- \$C9EF	1960	- \$FA90
90	- \$82DD	1970	- \$7450
100	- \$841B	1980	- \$FFFC
110	- \$1ABF	1990	- \$33D7
120	- \$CAFE	2000	- \$D3AE
130	- \$71F1	2010	- \$15DA





# Readers Data EXchange



140 - \$1A42	2020 - \$5191	780 - \$A012	2660 - \$52F8	1420 - \$8395	3290 - \$3E7C
150 - \$7048	2030 - \$2BDA	790 - \$B7F9	2670 - \$AA4D	1430 - \$E5B2	3300 - \$E1DE
160 - \$788B	2040 - \$B6E0	800 - \$8871	2680 - \$34F8	1440 - \$031E	3310 - \$B888
170 - \$FA70	2050 - \$7E7A	810 - \$E8D1	2690 - \$1382	1450 - \$6CEE	3320 - \$0FBF
180 - \$EA82	2060 - \$E6EC	820 - \$79F0	2700 - \$1D5F	1460 - \$A45A	3330 - \$0A59
190 - \$F704	2070 - \$7B35	830 - \$B5C4	2710 - \$74F9	1470 - \$951A	3340 - \$5FF7
200 - \$30DC	2080 - \$B1E3	840 - \$E0AC	2720 - \$E0CD	1480 - \$40DD	3350 - \$E5D9
210 - \$17C9	2090 - \$4D83	850 - \$0E7B	2730 - \$1D90	1490 - \$B183	3360 - \$2665
220 - \$B39C	2100 - \$DCAA	860 - \$B0F0	2740 - \$4BA8	1500 - \$CD79	3370 - \$E8B3
230 - \$BE6A	2110 - \$BFCB	870 - \$9E34	2750 - \$0801	1510 - \$C958	3380 - \$5F0D
240 - \$2FBE	2120 - \$99B0	880 - \$618C	2760 - \$2CA8	1520 - \$0A19	3390 - \$BC52
250 - \$FE0E	2130 - \$9BE5	890 - \$1801	2770 - \$E41E	1530 - \$32BA	3400 - \$DF3F
260 - \$FF3E	2140 - \$E645	900 - \$23BC	2780 - \$4AC0	1540 - \$6F5B	3410 - \$AE9C
270 - \$8321	2150 - \$FCA2	910 - \$1FAF	2790 - \$AB8E	1550 - \$0035	3420 - \$D649
280 - \$A903	2160 - \$18C1	920 - \$3D1A	2800 - \$6534	1560 - \$3C89	3430 - \$17F7
290 - \$10C1	2170 - \$7AD8	930 - \$A732	2810 - \$5258	1570 - \$AC12	3440 - \$9D1A
300 - \$7155	2180 - \$DD38	940 - \$A9C2	2820 - \$8B3C	1580 - \$C8F6	3450 - \$693F
310 - \$9383	2190 - \$AE53	950 - \$516C	2830 - \$40A5	1590 - \$E732	3460 - \$F830
320 - \$90CF	2200 - \$7F4C	960 - \$124B	2840 - \$F0E4	1600 - \$BB30	3470 - \$D06A
330 - \$2642	2210 - \$DDFA	970 - \$6966	2850 - \$1688	1610 - \$FA63	3480 - \$E987
340 - \$92C8	2220 - \$B5FE	980 - \$5319	2860 - \$91C6	1620 - \$D108	3490 - \$7816
350 - \$8E38	2230 - \$A4DB	990 - \$C11C	2870 - \$05AC	1630 - \$65C9	3500 - \$EC5B
360 - \$7FBD	2240 - \$A239	1000 - \$96D5	2880 - \$FB33	1640 - \$5A89	3510 - \$2109
370 - \$70C4	2250 - \$40F9	1010 - \$76F8	2890 - \$AE12	1650 - \$8D93	3520 - \$A280
380 - \$1914	2260 - \$EBA1	1020 - \$585B	2900 - \$59EB	1660 - \$50B1	3530 - \$8FD0
390 - \$378E	2270 - \$F09B	1030 - \$54DD	2910 - \$5F6B	1670 - \$A492	3540 - \$1605
400 - \$8F39	2280 - \$D074	1040 - \$F14B	2920 - \$2412	1680 - \$BC11	3550 - \$21D1
410 - \$83CC	2290 - \$C7E0	1050 - \$7474	2930 - \$6C38	1690 - \$6349	3560 - \$3035
420 - \$8838	2300 - \$4480	1060 - \$749C	2940 - \$BE02	1700 - \$58B8	3570 - \$0280
430 - \$2C79	2310 - \$FBF7	1070 - \$7B40	2950 - \$43DF	1710 - \$F850	3580 - \$CE58
440 - \$D2B1	2320 - \$1701	1080 - \$802D	2960 - \$B1F8	1720 - \$BE80	3590 - \$CA42
450 - \$5212	2330 - \$A9AF	1090 - \$29A2	2970 - \$6D18	1730 - \$46E4	3600 - \$3FC5
460 - \$B8C5	2340 - \$585C	1100 - \$DBE0	2980 - \$46AC	1740 - \$6D57	3610 - \$4FBC
470 - \$7340	2350 - \$2F6D	1110 - \$4726	2990 - \$EFE6	1750 - \$ADFB	3620 - \$0D86
480 - \$03DB	2360 - \$A802	1120 - \$BE3A	3000 - \$4761	1760 - \$4101	3630 - \$E4C8
490 - \$2BBD	2370 - \$4F6F	1130 - \$BFDf	3010 - \$5D37	1770 - \$1839	3640 - \$5046
500 - \$0C65	2380 - \$BAEC	1140 - \$3883	3020 - \$DF49	1780 - \$2335	3650 - \$B253
510 - \$0D45	2390 - \$70DD	1150 - \$0FC0	3030 - \$7AD5	1790 - \$BFB0	3660 - \$16B5
520 - \$8CF5	2400 - \$1A45	1160 - \$1BBA	3040 - \$8927	1800 - \$847F	3670 - \$BF89
530 - \$8C39	2410 - \$6242	1170 - \$6978	3050 - \$FA90	1810 - \$6099	3680 - \$4898
540 - \$7A23	2420 - \$34DB	1180 - \$1896	3060 - \$AC24	1820 - \$B392	3685 - \$8477
550 - \$E3A2	2430 - \$973B	1190 - \$257E	3070 - \$3642	1830 - \$665B	3690 - \$ED92
560 - \$4609	2440 - \$656B	1200 - \$9202	3080 - \$E96D	1840 - \$78EE	3700 - \$7436
570 - \$42FE	2450 - \$039A	1210 - \$CA2A	3090 - \$39B2	1850 - \$33D2	3710 - \$8E7A
580 - \$AB65	2460 - \$6E87	1220 - \$41E8	3100 - \$6E5A	1860 - \$56E4	3720 - \$0306
590 - \$1D4C	2470 - \$1F0E	1230 - \$31B2	3110 - \$A8AA	1870 - \$831E	3730 - \$1CC6
600 - \$9F39	2480 - \$ACE8	1240 - \$0A38	3120 - \$9F36	1880 - \$85F1	3740 - \$04F2
610 - \$3F3A	2490 - \$EFE0	1250 - \$ECC2	3130 - \$79FD	1890 - \$6F48	
620 - \$8D75	2500 - \$42EA	1260 - \$8D19	3140 - \$F05F		
630 - \$2708	2510 - \$CEB5	1270 - \$75C1	3150 - \$37E0		
640 - \$C929	2520 - \$2BA7	1280 - \$4722	3160 - \$BA11		
650 - \$D33C	2530 - \$572B	1290 - \$F890	3170 - \$7A5E		
660 - \$B35E	2540 - \$0E1F	1300 - \$FAC6	3180 - \$9C06		
670 - \$2452	2550 - \$80CC	1310 - \$2403	3185 - \$97EF		
680 - \$397F	2560 - \$5FCB	1320 - \$10C5	3190 - \$8DFD		
690 - \$8599	2570 - \$6E13	1330 - \$237B	3200 - \$E643		
700 - \$0602	2580 - \$95DB	1340 - \$A8D4	3210 - \$172F		
710 - \$AE4C	2590 - \$645D	1350 - \$9904	3220 - \$D7A2		
720 - \$A594	2600 - \$B1D5	1360 - \$26A1	3230 - \$7BBE		
730 - \$309A	2610 - \$3555	1370 - \$58CD	3240 - \$EBC9		
740 - \$55A5	2620 - \$89F4	1380 - \$0E2F	3250 - \$DC5F		
750 - \$0A71	2630 - \$D739	1390 - \$E15F	3260 - \$9D41		
760 - \$B297	2640 - \$BED7	1400 - \$08B1	3270 - \$9B38		
770 - \$4B21	2650 - \$C457	1410 - \$5995	3280 - \$C87B		

Francisco-Angel Garcia Moruja

Softkey for...

**Your Personal Net Worth v1.1**

Scarborough

If a reader merely wants to copy the program, use the Bitcopy option of *Copy II Plus* v5.x, 6.x or 7.x with the following parameters for the Auto Copy function, which are valid for all three versions.

T0-T2, SECTOR COPY  
SECTOR EDIT, TRACK 2, SECTOR 00.  
DOS 3.3, B7:08/0B/03/80  
T4-T10, SECTOR COPY  
T11, SECTOR COPY, 5C=D5  
RESTORE  
T12-T23, SECTOR COPY

## ■ Requirements

- Copy II Plus version 5.x, 6.x or 7.x
- 2 DOS 3.3 formatted disks

**1** Make a copy of the original disk using the Manual Sector Copy function of the Bitcopy option of *Copy II Plus*, copying tracks T0 to T2, T4 to T10 and T12 to T23. This is done so that tracks \$03 and \$11 keep their DOS 3.3 format.

**2** Using the sector editor with Patcher in Custom, change Check Address Epilog to NO. Copy track \$11, sectors 00,06-0F reading from the original disk and writing to the copy disk. This copies the VTOC and Directory.

**3** With the sector editor function in DOS 3.3, on track \$02, sector \$0C, change the bytes at \$B7-BA to 08 0B 03 80. This change blocks the protection of track \$03. Also on track \$11, sector \$00, change the byte at \$01 to 11 and the byte at \$40 to FE. This is done so that the Directory is pointed to track \$11 and so that sectors \$09-0F of track \$02 can be used.

**4** Use the Copy Files function to transfer all files from the copy disk to the second DOS 3.3 formatted disk.

**5** Finally, use the Change Boot Program function to change the boot name to "NW".

You may now boot the second disk because it is a normal DOS 3.3 disk.

**Greiner Wolfgang**

Softkey for...

## Crypt of Medea

Sir-Tec Software, Inc.

## ■ Requirements

- Apple II+, IIc, IIc
- One blank disk
- Super IOB v1.5

*Crypt of Medea* is a picture text adventure where you must escape from Medea's forbidden tomb. Overwhelming terror and ghoulish obstacles await you at every turn. The graphic routines are from Penguin software's *Graphics Magician*. For those of you with a mockingboard, the text will be spoken, but you don't need one to play.

There are two good things:

1. There is a backup on the other side of the disk.

2. There is a hint-and-answer sheet included with the manual, if you can't go further.

Now let's start with deprotecting the program. Before I started with the boot code tracing, I examined the disk with the *Linguist* from the CIA (you can also use *The Nibbler* from COMPUTIST #19). The raw nibble dump on track \$00 shows normal address, data headers and trailers on physical sectors: 00,0D,0B,09,07,05,03,01,0E,0C corresponding logical sectors: 00-09. The other tracks and sectors have a different address header.

There are two protecting schemes on the disk to prevent it from being copied with COPYA. First, normal DOS address headers starts with D5 AA 96 and they changed it to B5 AA 96 on every sector except track \$00, sector \$00-09. Secondly, they changed the place of the track and sector by the translation of the address field header. For example: D5 AA 96 FF FE AA AB AF AE FA FB, the translation of the code (encoded in 4&4) after the address field header will be: FE 01 0E F1, (FE=volume, 01=track, 0E=sectore, F1=checksum). On the Medea disk the code will be: FE 0E 01 F1.

Now let's start with the boot code tracing, which is my favorite method of deprotecting programs. If you do not wish to learn anything about boot code tracing this disk, skip straight to the end and copy it with *Super IOB*.

**1** We move the boot code from ROM to RAM, change its exit to jump to the monitor rather than the next stage.

Call -151  
8600<C600.C6FFM  
86F9:59 FF  
8600G

The drive will come on and soon you will get a beep as you jump to the monitor.

**2** Turn off the drive and see what we have.

COE8  
801L

Looking at location \$800 we find an 01, meaning only sector \$00 has been loaded into memory at \$800-\$8FF. Location \$84D-\$85C is the skewing table for translating logical into physical sectors. (*The practice of selecting the order of sectors on a diskette track to minimize access time due to rotational delay is called skewing.*) The indirect jump at \$836- JMP (\$3E) loads sector 00-09 at memory address \$B600-\$BFFF. The indirect jump at \$84A- JMP (\$8FD) goes to the second boot stage at \$B700. There we will put a break and examine the second stage.

**3** We move the first boot stage at memory address \$8000 and change the boot code at \$8600 to jump at this address.

86F9:01 80  
8000<800.8FFM  
800E:80  
804A:4C 59 FF  
8600G

Turn off the drive:

COE8  
B700L

A quick look at SB700 shows an IOB (Input/Output Block) beginning at \$B704-\$B714. At SB700 there is a jump to SB744 which sets the reset vector at location \$AC67. Afterwards, T:01/S:0F-00 will be loaded at \$800-\$17FF and T:02/S:0F-08 at \$1800-\$1FFF. At SB7A1 there is a JSR \$1D00 which is responsible for loading: T:22/S:02-0B at \$6000-\$69FF T:12/S:00-01 at \$2000-\$21FF.

Control returns then to the second boot stage. The code at \$B7A4-\$B7C3 loads the picture of the Medusa. Thereafter, a couple of other tracks and sectors will be loaded. We are not finished yet.

There is still a little more work to do. The code at \$B7DF-\$B7EF loads the memory wipe out routine to memory address \$AC00-\$ACFF (T:00/S:0F). At \$B7F3 there is a jump to \$1630.

**4** To look at this code make the following changes:

B7F4:59 FF  
B700G  
1630L

At location \$1634-\$1643 there is a second IOB. Another couple of tracks and sectors will be loaded.

**5** Now we examine the code at \$16D5-\$16EF which is as follows:

```
16D5 LDA μ$3F
16D7 PHA      Start address of main
16D8 LDA μ$FF program minus 1.
16DA PHA      $4000-1 = $3FFF
16DB LDA μ$B4
16DD STA $163D buffer high ($B400)
16E0 LDA μ$00
16E2 STA $1638 track $00
16E5 LDA μ$0E
16E7 STA $1639 sector $0E
16EA JSR $165D loads T:00/S:0E at $B400
16ED JMP $B400 third boot stage
```

**6** Now we take a look at the code at \$B400:

16EE:59 FF  
1630G  
B400L





At SB403-SB413 there is a third IOB. The code that is important to us starts at \$B43E and looks as follows:

```
B43E LDA $C0E9 turn motor on
B441 LDA $05 counter
B443 PHA
B444 JSR $B48A nibble counter
B447 BCS $B44D nibble counter failed or
B449 CMP $03 test if correct read
B44B BEQ $B456 yes no
B44D PLA get counter
B44F SBC $01 decrement counter
B44E SEC
B451 BNE $B443 next try else
B453 JMP $AC67 memory wipe out
B456 PLA
B457 LDA $C0E8 turn motor off
```

To defeat the nibble counter, I only place a JMP \$B45A instruction instead of the LDA \$C0E8 (turn motor on). This causes the code to continue after the instruction "turn motor off". The exit to the main program is at \$B468-RTS which pulls the two bytes \$3F and \$FF out of the stack. If we put the two bytes together, the start address will be \$4000 (\$3FFF+1).

### Summary

- 1 Install the controller program at the end of this article into *Super IOB v1.5*.
- 2 Copy the *Crypt of Medea* disk with *Super IOB*. Ignore the clacking noise of the disk drive on track \$00, sectors \$0C-0F, everything will be copied correctly.
- 3 Make the following sector edits to the disk.

This will bypass the nibble counter.

Trk	Sct	Byte(s)	From	To
\$00	\$0E	\$3E-40	AD E9 C0	4C 5A B4

This will point the reset vector to \$FF59 instead to \$AC67.

Trk	Sct	Byte(s)	From	To
\$00	\$01	\$45	67	59
		\$4A	AC	FF

This will read the right address header on your copy.

Trk	Sct	Byte(s)	From	To
\$00	\$03	\$55	B5	D5

This will write the correct address header during initialization if you chose the Save Game option, and the right track-and sector order.

Trk	Sct	Byte(s)	From	To
\$00	\$06	\$7A	B5	D5
		\$8E	3F	44
		\$93	44	3F

This will read the right track and sector from the sector header.

Trk	Sct	Byte(s)	From	To
\$00	\$07	\$EE	2D	2E
\$00	\$08	\$2F	2E	2D

### Note

You can reset into the monitor at every moment, when you play the game. To continue enter:

```
C050
C057
4000G
```

On my Apple IIe, the letters changed to lower case and the inverse direction status doesn't work correctly.

### Controller

```
1000 REM CRYPT OF MEDEA
1010 TK = 0:LT = 35:ST = 0:CD = WR
1015 ONERR GOTO 550
1020 T1 = TK:GOSUB 490
1025 IF TK = 0 AND ST < 10 THEN GOSUB 430:GOSUB
100:ST = ST + 1:IF ST < DOS THEN 1025:REM
T:00/S:00-09
1030 RESTORE:GOSUB 190:POKE 48622,45:POKE
48687,46:REM read changed track and
sector
1035 GOSUB 430:GOSUB 100:ST = ST + 1:IF ST < DOS
THEN 1035
1040 IF BF THEN 1050
1045 ST = 0:TK = TK + 1:IF TK < LT THEN 1035
1050 GOSUB 230:POKE 48622,46:POKE 48687,45
1055 GOSUB 310
1060 GOSUB 490:TK = T1:ST = 0
1065 GOSUB 430:GOSUB 100:ST = ST + 1:IF ST < DOS
THEN 1065
1070 ST = 0:TK = TK + 1:IF BF = 0 AND TK < LT THEN
1065
1075 IF TK < LT THEN 1020
1080 HOME:PRINT "DONE"WITH"COPY":END
5000 DATA ^181,170,150
5010 DATA ^11^CHANGES
5020 DATA ^0,14,62,76,0,14,63,90,0,14,64,180
5030 DATA ^0,1,69,89
5040 DATA ^0,1,74,255
5050 DATA ^0,3,85,213
5060 DATA ^0,6,122,213
5070 DATA ^0,6,142,68
5080 DATA ^0,6,147,63
5090 DATA ^0,7,238,46
5100 DATA ^0,8,47,45
```

### Controller Checksums

1000	- \$356B	1075	- \$0B19
1010	- \$5824	1080	- \$2E19
1015	- \$FBCC	5000	- \$A81E
1020	- \$7CF0	5010	- \$617F
1025	- \$E1D8	5020	- \$8165
1030	- \$3C58	5030	- \$4B3E

1035	- \$295A	5040	- \$279A
1040	- \$9466	5050	- \$DB08
1045	- \$7474	5060	- \$BFAC
1050	- \$31F3	5070	- \$F92F
1055	- \$811A	5080	- \$9B2A
1060	- \$65D2	5090	- \$0DA9
1065	- \$69D5	5100	- \$EF12
1070	- \$7BCB		

Marshall P. Brown

Sofiskey for...

## Digital Paintbrush System

The Computer Colorworks

### Requirements

- Super IOB v1.5
- Two blank disks
- Sector Editor
- An Apple II or II Plus with 64K of RAM

When I first saw the advertisement for the *Digital Paint Brush System* I was immediately impressed with everything but the price, \$299 plus shipping, and they stated that the utilities were unprotected. After laying down my money, I discovered that, true, the utilities were unprotected but the programs were protected.

Being rather new to the deprotection game and having seen only a couple copies of *COMPUTIST*, I was unable to do much about it. As it turned out *EDD III* would copy the programs, and true enough the company does provide back-ups with the original purchase so there is no extra charges or waiting for the mail. I was somewhat discouraged, however, by the fact that each of the back-ups had one bad program on them and when I finally crashed my duplicates, I decided it was time for a crack.

As a first shot, I used a *Wildcard* to capture the RWTS (turned out to be unnecessary) and *Super IOB* with the swap controller. The results looked fairly normal but refused to run, which is also fairly normal.

Falling back to *Disk Edit*, I took a close look at the original disk and found that the address and data trailers had been altered from the normal DE AA EB to AA AA EB. It looked as if all I had to do was change the almost standard DOS back to normal, but I recalled that the program not only writes to a standard disk but additionally writes configuration information to the protected disk. This means that the program must be able to change the trailers on the fly, and that simply changing the DOS wouldn't stop the program from changing them back.

After sector editing the DOS trailers on my duplicate, it was time to get out *Locksmith 6.0's Auto Bootcode Tracer*. With the ABT active, I booted the *Digital Paintbrush* disk.



I had set the ABT to stop when there was any attempt at writing to \$B991, one of the DOS locations containing the trailer information. I soon found that at one point a 4C was being written to \$B991. I took note of the surrounding code and then did a search of the disk using *Copy II Plus*. The offending code was located in FLYING COLORS PRO.1 and duly changed back to a DE with the sector editor.

The next step was to examine the BASIC programs on the disks for any further protection. Most of the BASIC programs have a short machine code program hung on the end which loads with the BASIC program and then is located by PEEKing at \$AF and \$BO. For the most part, these hang on programs set the & vector and perform some other housekeeping.

In examining the BASIC programs you will find a variable, DC, which needs to be changed to 00. You will also find, in one or two lines, a POKE to \$B991, a CALL to \$B991 and a second POKE to \$B991. The effect of this is to change the AA to 60 (RTS), CALL it, and then restore it to AA. We will change the CALL to \$B99F which already is a RTS.

The BASIC programs on the second *Digital Paintbrush* disk have altered DOS commands. This need not affect the deprotection but should you want to use a fast DOS they will need to be changed back to the correct commands using a program such as *DOS Boss* from Beagle Brothers.

Standard	Changed To
INIT	FRESH
RUN	MUSH
DELETE	XINK
CATALOG	LOOK
FP	LIST
BSAVE	BLOO
BLOAD	VELDT
BRUN	HILDA
All others	XX

A search and replace utility such as found in Beagle Brother's GPLLE is a great time saver in normalizing these changes. Since my main intention was to produce a working back-up, I have not worked out the details of completely normalizing the programs.

**1** Type in the controller as shown and save it as DIG PAINT.CON.

### Controller

```
1000 REM DIG PAINT.CON
1010 TK=0:LT=35:ST=15:LS=15:CD=WR:FAST=1
1020 GOSUB 170:RESTORE:GOSUB 490:GOSUB 610
1030 GOSUB 230:GOSUB 490:GOSUB 610:IF PEEK(TRK)=LT THEN 1050
1040 TK=PEEK(TRK):ST=PEEK(SCT):GOTO 1020
1050 HOME:PRINT "COPY DONE":END
1060 DATA "170,170,170,170"
```

### Controller Checksums

1000	- \$356B	1040	- \$D6FC
1010	- \$2544	1050	- \$16BB
1020	- \$579A	1060	- \$F867
1030	- \$6203		

**2** Using *Super IOB v1.5* copy both disks.

**3** Sector edit both disks as follows:

Trk	Sct	Byte(s)	From	To
\$00	\$02	\$9E	AA	DE
	\$03	\$35	AA	DE
		\$91	AA	DE
	\$06	SAE	AA	DE

**4** Sector edit disk 1 as follows:

Trk	Sct	Byte(s)	From	To
\$0C	\$06	\$1C	\$4C	\$DE

**5** Boot a standard DOS 3.3 disk and load SLIDE PROJECTOR.

**LOAD SLIDE PROJECTOR 2.0**

**6** In line 100, change DC from -52 to -00. In line 180, change 126 in two places to 000, and change 47505 to 47519.

**7** Unlock, save and lock SLIDE PROJECTOR.

**UNLOCK SLIDE PROJECTOR 2.0**

**SAVE SLIDE PROJECTOR 2.0**

**LOCK SLIDE PROJECTOR 2.0**

**8** Load TEXT EDITOR.

**LOAD TEXT EDITOR 1.0**

**9** In line 70, change DC from -52 to -00. In line 100, change 126 in two places to 000, and change 47505 to 47519.

**10** Unlock, save and lock TEXT EDITOR.

**UNLOCK TEXT EDITOR 1.0**

**SAVE TEXT EDITOR 1.0**

**LOCK TEXT EDITOR 1.0**

**11** Load SLIDE WRITER.

**LOAD SLIDE WRITER 1.0**

**12** In line 80, change DC from -52 to -00. In line 150, change 126 in two places to 000, and change 47505 to 47519.

**13** Unlock, save and lock SLIDE WRITER.

**UNLOCK SLIDE WRITER 1.0**

**SAVE SLIDE WRITER 1.0**

**LOCK SLIDE WRITER 1.0**

**14** From disk 2 load PACKER.

**LOAD PACKER**

**15** In line 500, change DC from -52 to -00. In line 600, change "X" from 47505 to 47519, in two places change 126 to 000.

**16** Unlock, save and lock PACKER.

**UNLOCK PACKER**

**SAVE PACKER**

**LOCK PACKER**

The *Digital Paintbrush System* will now run and is *COPYA*-able.

David Goforth

Softkey for...

**Test Maker**

Bertamax

### Requirements

- Test Maker disk
- One blank disk
- Super IOB 1.5
- Controller listed below
- Sector Editor
- DOS copier
- Optional: Copy II Plus v7.4

*Test Maker* is a program that allows tests to be entered in the computer in several different formats (multiple choice, true/false, fill in, matching, etc.). It then prints out the test in whatever order the person desires and will also print out an answer key for each test.

A friend recently brought this to me and asked if I could remove the copy protection, so that he could take the tests without printing them out on paper. After looking through several copy programs for parameters (to no avail), I decided to start trying to copy it. Disk copiers could not read it and bit copied copies would not boot.

So I got out *Copy II Plus 7.4* and using the sector editor (patched mode) found that it contained a normal catalog track but no programs resided there. Huh? Looking through the DOS tracks (0-2), I found that it booted an Applesoft program called hello. Scanning the disk for this file revealed the true catalog on track \$0F starting with sector \$0F along with the following files.

Type	Filename	ADDR	LENGTH
A	HELLO		
B	STR	A\$300	L\$C8
B	OBJ	A\$9300	L\$2E3
B	OBJ IIE	A\$B500	L\$275
B	DDMOVER	A\$8700	L\$AFA
A	TEST MAKER		
A	TEST MAKER 2		
T	PRINT CONSTANTS		





Print constants is a sequential file which uses the following variables:

Variable	Function
SN	printer slot
PD\$	pre-print code
PL\$	letter quality
TQ	page length
PCS	normal print
TP	text length
ND	number of disk drives (1-2)

I also found that part of the disk was in normal DOS format (including a real catalog on track \$11 - although empty) and the rest contained altered epilogues. So basically it contained a slightly modified DOS (so as to find the catalog on track \$0F) and to read the altered epilogues. Using *Super IOB* (ignoring errors and data epilogues), I copied the disk. Then, returning to the sector editor I copied track \$0F, sectors \$0F and \$0E to track \$11, sectors \$0F and \$0E. I changed the first two bytes to reflect the next catalog sector as track \$11 rather than \$0F.

After that I copied a standard DOS onto the disk so that it would boot with the standard epilogues. It booted, loaded hello, then hung. This told me that there was more protection within the actual programs, so I booted a normal DOS disk and loaded the hello file without running it.

Listing the hello file showed me all the protection methods used. They were all in the form of POKES which changed between two catalog tracks and altered normal epilogues. All I had to do was delete six lines and try once again. It booted, loaded hello, loaded *Test Maker* (for Apple II Plus), and hung. Repeating the process I used for the hello file, I loaded the *Test Maker* file and again found two lines where the POKES were used. I deleted them and saved *Test Maker*. It worked. You now have a deprotected copy to change as you please.

**1** Install the controller and lines 215 and 218 into *Super IOB*. (This is a standard ignore controller with line 1015 added. It will work without the ignore data epilogues but it takes much longer and sounds terrible!)

**2** Copy *Test Maker* onto a blank formatted disk using *Super IOB*. Put away the original, we're done with it.

**3** Using something like *Copy II Plus 7.4*, copy a normal DOS to the copy.

**4** Make these edits. Be careful, you'll be writing the sectors back to a different location then where you read them.

Trk	Sct	Byte(s)	From	To
\$0F	\$0F	\$01	0F	11

Write it back to track \$11, sector \$0F.

Trk	Sct	Byte(s)	From	To
-----	-----	---------	------	----

\$0F \$0E \$01 0F 11

Write it back to track \$11, sector \$0E.

**5** Boot a normal DOS disk (not *Test Maker*).

**6** Insert the *Test Maker* copy and remove the altered epilogue POKES and track \$0F catalog POKES from HELLO and TEST MAKER.

### LOAD HELLO

65  
352  
353  
620  
621  
622

### SAVE HELLO

### LOAD TEST MAKER

28  
37

### SAVE TEST MAKER

There are two similar files on the disk (one for Apple II Plus's and the other for Apple IIe's.) The files for Apple IIe's did not contain any POKES that I could find. If you have problems booting while using an Apple IIe, then load the file *Test Maker 2* and list it, looking for any POKES in the range of 400000 to 480000, these are the lines you want to delete.

That's it! You now have a deprotected copy of *Test Maker*. Files may be saved on the original disk on the normal catalog track (\$11). If you have any files or tests there, they can be copied with any file copier to your new disk.

## Controller

```

215 REM IGNORE DATA EPILOGUES
218 POKE 47415,0: POKE 47416,189: POKE
47417,140: POKE 47418,192: POKE 47419,16:
POKE 47420,251: POKE 47421,201: POKE
47422,170: POKE 47423,76: POKE 47424,158:
POKE 47425,185: RETURN
1000 REM TEST MAKER CONTROLLER
1010 TK = 0: LT = 35: CD = WR: MB = 151: ONERR GOTO
550
1015 GOSUB 215
1020 ST = 0: T1 = TK: GOSUB 490
1030 GOSUB 430: GOSUB 100: ST = ST + 01: IF ST <
16 THEN 1030
1040 IF BF THEN 1060
1050 ST = 0: TK = TK + 01: IF TK < LT THEN 1030
1060 TK = T1: ST = 0: GOSUB 490
1070 GOSUB 430: GOSUB 100: ST = ST + 01: IF ST <
16 THEN 1070
1080 ST = 0: TK = TK + 01: IF BF = 0 AND TK < LT THEN
1070
1090 IF TK < LT THEN 1020
1100 HOME : AS = "ALL DONE" : GOSUB 450: END

```

## Controller Checksums

215	-	\$CF09	1040	-	\$3D68
218	-	\$1593	1050	-	\$9E1A
1000	-	\$6219	1060	-	\$04BA
1010	-	\$1BFF	1070	-	\$4427
1015	-	\$54AD	1080	-	\$84C0
1020	-	\$03B6	1090	-	\$A792
1030	-	\$5D2F	1100	-	\$A102

One more word for those out there who want to see more educational softkeys. Some of the reasons that you don't see as many as you like are:

1. They're not as popular as games, thus not as many users are trying to copy them.

2. Of those that I have tried to copy, the protection is normally far more complex than that of most games on the market. The manufacturers seem to go out of their way to insure that the protection is difficult to find and remove. The people who could copy it usually are not those who buy these programs, but rather are parents and teachers.

3. The only suggestion I can offer is that COMPUTIST is your best bet at finding ways to copy programs. You may also try to find other subscribers within your area (usually in computer users groups) who know more about copy protection than you (or have more time). They may be willing to try to copy these for you, usually just for the chance to further their abilities. Some may charge a minimal fee (if any) for their time, but it may well be worth it.

Ron Balewski

Softkey for...

## Arctic Antics

Epyx

### Requirements

- Arctic Antics disk
- Blank disk for copy
- COPYA or equivalent
- Sector editor

A friend recently got *Arctic Antics* by Epyx. Naturally, he wanted to make a back-up copy like everyone suggests. And, naturally, the disk was protected. He tried several things to crack the disk without success. So, I got the chore.

The first thing I tried was looking up past softkeys for Epyx software. I found that Brian A. Troha's softkey for *Destroyer* in COMPUTIST #49 was close, but did not cut it. At least it was a start.



I started *Locksmith's* tracer/simulator feature and followed the boot sequence. All seemed fine until execution entered page \$0C. Then, the computer reset via a jump to (\$FFFC). Something in this page was doing the disk check. So, I paused the simulator, entered the monitor, and disassembled \$0C00 - \$0CFF to the printer. Now for the fun — slugging through undocumented assembly code!

What I found after a bit of head-scratching was the disk decoding routine Mr. Troha spoke of except this one seemed souped up. Listing 1 contains selected parts of the routine — documented by yours truly for your reading pleasure. In a nutshell, here is what it does.

First it writes a STA \$C009 instruction to location \$082A. I suppose they wanted to keep us from bypassing this routine completely by having it place one important instruction somewhere else. Pretty sneaky!!

Next, it looks for various nibbles on the disk. All the while, it seems to keep count of how many nibbles it reads between what it expects. If too many unmatched nibbles go by, it decrements an error counter and restarts the protection check. After 10 errors, when the countdown reaches zero, \*\* CRASH \*\* it goes into reset!

Eventually, it finds an EE. When it does, it loads the next eight nibbles into memory locations \$F7 to \$F0. It then seems to use the last three as decryption keys to decode three pages of memory, pages \$09, \$0A, and \$0B.

Finally, it jumps to \$082A (the instruction it set up earlier) to continue with the load.

Now that I knew exactly what the protection did, I decided it was time to come up with a way around it. I knew I could not just bypass this routine because of the infernal decryption it does. Without that, I knew the program didn't have any chance of continuing. I then thought about NOPing all the weird disk accesses and just going to the decryption section. But there's a problem — the decryption key comes from all those disk accesses! Okay, maybe I can NOP the disk accesses and load the eight-byte table manually. But with what?

I tried booting *Arctic Antics* again under the *Locksmith* Simulator's control. I hoped to pause immediately after the small table was loaded. Unfortunately, it didn't work. For some as yet unknown reason, the necessary \$EE never showed up with the simulator in control.

Finally, out of desperation, I booted the disk normally and quickly did a hard **[RESET]**. I didn't expect the values to still be in memory. I was sure Epyx would erase them as soon as they could. But I had to try something.

Well, the values were indeed still there. I then simply wrote a patch to stuff the values into the proper locations and put it right on top of the disk-access stuff. After saving the patch to the clone disk, I crossed my fingers and booted it. It worked!

There was still one slight problem. The copy I was working on was made with *Copy II Plus's* sector copier in DOS 3.3 patched mode. Because there was something odd about the epilogs, *Locksmith's Fastcopy* wouldn't work.

Fortunately, this was more of an inconvenience than a problem. I just patched DOS to ignore errors with the DOS Alterer and made a normalized copy with Franklin's standard Copy program (*COPYA* to you Apple people).

Here's the cookbook method:

- 1** Boot DOS 3.3.
- 2** Set DOS to ignore errors. I like using the DOS Alterer on the COMPUTIST Starter Kit disk.
- 3** Insert your DOS disk and RUN COPYA.
- 4** Make the following edits:

Trk	Sct	Byte(s)	From	To
\$00	\$0B	\$18-\$86	A9 0A 85 FC	A9 FC A5 F0
			A6 2B BD 89	A9 EE 85 F1
			C0 BD 8E C0	A9 EE 85 F2
			A9 80 85 FD	A9 FC 85 F3
			C6 FD F0 7C	A9 E7 85 F4
			?	A9 EE 85 F5
			?	A9 FC 85 F6
			?	A9 E7 85 F7
			?	All EA's
			?	up to and
			FB 99 F0 00	including
			EA 88 10 F4	Byte 86

That's it! *Arctic Antics* can now be copied with any fast copier. So put your original in the archive and don't worry about a crash!

### Listing 1

```

LDA #88B      Store
STA $082A    STA $C009
LDA #809     instruction
STA $082B    at $082A
LDA #8C0     instruction
STA #882C    instruction
- - -
LDA #0A      Set error countdown
STA $FC      to 10
- - -
LDY #810     Must find $EE within $10
BIT #80      bytes
$0C6C LDA $C08C.X Read byte from disk
BPL $0C6C
DEY          Dec counter
BEQ $0CA8   If 0, call it an error
CMP #8EE    Find the $EE?
BNE $0C6C   If not, try again
NOP
NOP
LDY #807     Load 8 bytes

```

```

$0C7C LDA $C08C.X Read a byte
BPL $0C7C
STA $0F0.Y   Stash it
NOP
DEY          Loop
BPL $0C7C   till done
LDX #802
LDA #800     Word at $F8 - $F9
TAY
STA $FB      is set to point
LDA #809     to memory
STA $F9      page $09
$0C93 LDA $F0.X Get decryption code
EOR ($F8).Y Decrypt a byte
STA ($F8).Y Put it back
DEY          Full page done?
BNE $0C93   If not, keep looping
INC $F9     Point to next page
DEX         next decryption byte
BPL $0C93   Loop for all pages
LDX $2B
LDA $27
JMP $82A    Instr inserted earlier

```

This is the error-counting time bomb routine.

```

0CA8 DEC $FC      Dec timebomb clock
BEQ $0CAF      Boom?
JMP $0C24     Not yet. Try again.
$0CAF INC $03F4  ** BOOM **
JMP ($FFFC)   Crash with reset

```

I recently came across a couple novel programs that were, of course, copy-protected. Since they weren't listed anywhere, I had to deprotect them myself. Here are the softkeys in case you should find these strange but enjoyable works:

### Softkey for...

**Paper Models —  
The Christmas Kit**  
Activision

### Requirements

- Paper Models disk
- One blank disk
- Fast copier (such as *Locksmith*)
- Sector Editor (such as *Copy II Plus*)

This rather unusual program will print cut-out paper model patterns on your printer. There are some interesting ones, such as a train engine and a few cars.

The protection is standard Activision. Deprotection was quite easy. Just use the normal Activision method of:

**1** Fast copy the disk with *Locksmith*, *Disk Muncher*, etc.

**2** Search for A9 56 85 FD A9 ... 38 2A 25 FC. (the "..." means all of the bytes in between.)



**3** Change the string to EAEAEA ... EA9 FF

I found this string starting at track \$12, sector \$03, byte \$58. Try looking there before going through the trouble of searching.

Softkey for...

## Railroad Works

CBS Software

### Requirements

- The Railroad Works disk
- Two blank disks (one temporary)
- EDD IV or IV Plus
- DOS 3.3 file copier such as FID
- DOS Alterer (from COMPUTIST Starter Kit)

This is a fairly old (1984) but really nifty model train simulator. Being a model railroad buff, I enjoy this type of program. This one is also light-years ahead of the new *Run Your Own Train* by Abracadata (By the way, I haven't completely cracked the Abracadata disk yet. Any help will be appreciated).

The copy protection wasn't all that difficult to break, once I figured out what they were doing. However, I used a couple of new methods that aren't mentioned in COMPUTIST too often. Maybe I can give you some ideas to help you crack some of your own software.

The only changes to this disk were changing the address headers to AA D5 AB (normal is D5 AA 96) and the data headers to AA D5 EB (normal is D5 AA AD) on all but the first sector. A quick scan with *Copy II Plus*' nibble editor brought forth this information.

I could have normalized this disk with *Super IOB*, but I wanted to do something different. After reading Stanley Planton's article on *EDD IV* in COMPUTIST #51, I thought I'd try fixing this disk with it.

After digging out and reading *EDD*'s manual, I'd like to clarify something in Mr. Planton's article. He's completely accurate in HOW to enter programming data into *EDD*. However, he doesn't really explain WHY it must be done that way. I find that knowing why helps in this case. The first parameter you set, 00, is really a pointer to the first instruction. Mr. Planton uses 20, but you can put the code anywhere in the 255-byte buffer — even starting at 01. All you have to do is set location 00 to wherever you put the code.

Now, on to the crack!

**1** Start *EDD IV* and enter the following PREWRITE program:

```
31 AA D5 AB 36 D5 AA 96 21 31 AA D5 EB 36
D5 AA AD 21 10
```

Now, use *EDD*'s standard nibble copy to copy the disk and fix all the headers.

**2** At this point the disk is almost normal, but there is still something amiss with (probably) the epilogs. We'll take care of that by copying the files onto a truly normal disk. So, get another blank disk and initialize it to DOS 3.3 with:

### INIT HELLO

**3** Now, boot your COMPUTIST Starter Kit disk (another quality product that I don't see used often enough) and run the DOS Alterer program. Change DOS to ignore errors by selecting Option 3 and typing I. You should now be able to CATALOG your *EDD*'d *Railroad Works* disk with Option 7. If you get a catalog, you're home free. If not, you did something wrong with *EDD IV*.

**4** Reset out of DOS Alterer and BRUN your File Utility Program. I used Franklin's FUD, but Apple's FID is about the same. Have FID/FUD/WHATEVER copy all the files from the *EDD*'d *Railroad Works* disk to the INIT'd blank disk.

You're done. Finished. Happy railroading!

Steve Cook

### A fix for Locksmith Fast Copy

Thanks for the postcard concerning the *Locksmith Fastcopy* article by William O. Romine in COMPUTIST #50. I have found the error. Where he says in column one: *SAVE FC, A\$2000, L\$18FD*, he meant: *SAVE FC, A\$2002, L\$18FD*.

J.D.Tischer

How about a way for us underprivileged Apple IIgs owners to reset into the monitor? P.S. On *Copy II Plus* v 8.2, use *Music Construction Set* to copy for *Marble Madness*!

**I** For the IIgs owners who have upgraded to the new 2.0 ROMs, there is a hidden classic desk accessory (CDA) that allows you to enter the monitor. In order to install the new CDA, you should enter the monitor before running any protected programs and press "#, [RETURN]". This will turn on two hidden CDAs, *Memory Peeker* and *Visit Monitor*. Thereafter press [ESC] to go to the Desk Accessories menu. Select "Visit Monitor" and there you are. Use [Y] to exit.

For those who have not upgraded, what are you waiting for?.....RDEXed

Where in the USA is Carmen San Diego Broderbund  
Wings of Fury Broderbund  
Word Attack Davidson  
Work Force II Core Concepts  
The Works First Star Software

# MOST WANTED Softkeys

- ABM Muse
- Accolade Comics Accolade
- Agent U.S.A. Scholastic
- Airheart Broderbund
- Algeblaster Davidson & Associates
- Algebra I Intelligent Tutor
- Apple Super Pilot ?
- Artificial Intelligence Scholastic
- Balance of Power Mindscape
- Bandits Sirius Software
- Bank Street Filer Broderbund
- Bank Street School Filer Sunburst Communications
- Barron's Computer SAT ?
- Battlegroup SSI
- Battlezone Atarisoft
- Brain Bank The Observatory
- Burgertime ?
- Calendar Crafter IIgs MECC
- Captain Goodnight ?
- Certificate Library Vol. 1 ?
- Championship Baseball ?
- Chuck Yeager's Advanced Flight Trainer Electronic Arts
- Colossus IV Firebird
- Creature Venture Softsmith Co.
- Cross Clues Science Research
- Cross Country Rally Softsmith Co.
- Crossword Magic Mindscape
- David's Midnight Magic ?
- DB Master V4.0 Stoneware
- Deathlord Electronic Arts
- Deep Space Sir Tech
- Dome Bookkeeping Systems Dome Accounting
- Fay: The Masked Woman Didatech Software
- Fay's Word Rally Didatech Software
- Fun Bunch Unicorn
- Galaxian Atarisoft
- Garfield Deluxe Edition DLM
- Gemstone Healer SSI
- GoldFinger Mindscape
- GradeBusters 1-2-3 Gradebusters
- Gutenberg Jr. & Sr. Micromation LTD
- Handicapping System Sports Judge
- J & S Grade Book J & S Software
- Jigsaw Microfun
- Joust Atarisoft
- Lazar Maze Avant-garde
- Legacy of the Ancients Electronic Arts
- Marble Madness GS Electronic Arts
- Mathblaster Davidson & Associates
- Maxi Golf Thunder Mountain
- Micro League Baseball Micro-league Sports
- Microzine #24-25 Scholastic
- Mr. Do Datasoft
- Mr. Pixel's Cartoon Kit Mindscape
- Ms. Pac-Man Atarisoft
- Never Ending Story Datasoft
- Odin Odesta
- Peeping Tom Microlab
- Pensate Penguin
- Personal Finance Manager (PFM) Apple Computer
- PFS File & Report IIgs Software Publishing Corp.
- Prime Plotter Primesoft Corp.
- Print Master Uniston World
- Pro-Football Sports Judge
- Publisher Springboard
- Quiz Castle Didatech Software
- Rescue On Fractalis Epyx
- Ruski Duck Softsmith Co.
- Scrabble Electronic Arts
- Snoggle Broderbund
- Space Eggs Sirius
- Space Journey Mindscape
- Stellar 7 Penguin Software
- Success with Typing V1.2 Scholastic Software
- Super Factory Sunburst
- Think Tank Living Video
- Tower of Myrargen II (IIgs) ?
- Toy Shop Broderbund
- Type Broderbund
- Ultima V Origin Systems
- Visiblend Microlab
- Wheel of Fortune ?

# COMPUTIST back issues

To order, Use the back issue order form on page 42

**54 March 1988** ■ **Features:** •A Utility to Save the Lower 8 Pages of Memory •Bard's Tale Effects Locator •How to Capture Phantasie Screen Maps •Alternate Reality Character Editor •Updating the ProDOS Block Editor •Loading Flashcalc onto your RAMcard •A Copy-protection Scheme for ProDOS •The Product Monitor •Autoduel Car Editor ■ **Softkeys:** •Alphabet Sequencing •Animal Alphabets and Other Things •Arctic Antics •The Boars' Store •The Boars Tell Time •Career Focus •Castle Wolfenstein •Charlie Brown's 123's •Charlie Brown's ABC's •City Country Opposites •Coveted Mirror •Create With Garfield •Crypt of Medea •Customized Alphabet Drill •Customized Flash Spelling •Dig Dug •Digital Paintbrush System •Estimation •Fay: Word Hunter •Fix It •Focusing on Language Arts •Fundamental Capitalization •Fundamental Punctuation Practice •Fundamental Spelling Words in Context •The Hobbit •Homonyms in Context •Individualized Study Master •Inside Outside Shapes •Inside Outside Opposites •Leisure Suit Larry in the Land of the Lounge Lizards •Master Diagnostics II & II+ •Mastertype v2.1 •Mathematics Series •Mr. and Mrs. Potato Head •Paper Models - The Christmas Kit •Peanuts Math Matcher •Peanuts Maze Marathon •Peanuts Picture Puzzlers •Perry Mason: The Case of the Mandarin Murder •Railroad Works •Random House Library •Management Programs •Rocky's Boots v.4 •Sensible Speller •Snoopy's Reading Machine •Snoopy's Skywriter Scrambler •Snoopy to the Rescue •Snoopy Writer •Spelling Demons •Stock Market Simulation •Story Builder •Story Starter •Studio II •Test Maker •Think Quick v1.0 •Tournament Bridge •Tutorial Comprehension •Typing is a Ball, Charlie Brown •Under Fire •Word Blaster •Word Count •Word Mount •Your Personal Net Worth **IBM** ■ **Feature:** •Flight Simulator RGB Modifications

**53 March 1988** ■ **Softkeys:** •2400 A.D. •Age Of Adventure •Apple's Core II •Arcade Boot Camp •Arctic Fox •Aztec •Ballblazer •Bard's Tale IIgs •Blue Powder Gray Smoke •California Games •Championship Wrestling •Colonial Conquest •Comprehension Skills I,II •Conquering Whole Numbers •Coordinate Math •Countdown To Shutdown •Dataquest: The World Community •Destroyer •Dream House •Dream Zone •Earth Orbit Station •Equation Math •Forecast: Your At-Home Weather Station •Fraction Concepts Inc •Fraction Munchers •Fraction Practice Unlimited •GBA Championship Basketball •Genesis •GFL Football •Ghost Rider •Goonies •Grade Manager v2.3 •Great American Cross-country Road Race •Hardball IIgs •Ikari Warrior •Jenny's Journeys •Kid Niki Radical Ninja •Kung-Fu Master •Learning To Tell Time •Leisure Suit Larry •Let's Learn About Money •Let's Learn About The Library •Letters For You •Lords Of Conquest •Magic Spells •Math Blaster •Money Works •Maps & Globes: Latitude & Longitude •Marble Madness •Microzine 18,19,20,21,22,23 •Mist

•Morning Star Math •Movie Monster Game •Multiplication Puzzles •Multiscribe v3.0c •Murder On The Mississippi •Music Made Easy •Mystery Sentences •Number Munchers •Numbers Count •Odell Lake •Operation Frog •Opposites Attract •Oregon Trail v1.4 •Phonics Prime Time: Blends & Digraphs •Phonics Prime Time: Vowels I, II •Puzzles & Posters •Quotient Quest •Reader Rabbit •Reading Style Inventory •Realm Of Impossibility •Sesame Street 'Crayon' series •Shanghai •Sons Of Liberty •Space Quest v2.2 •Story Book: Pixelworks •Story Tree •Subtraction Puzzles •Super Huey •Super Wordfind •Tass Times In Tonetown •Those Amazing Reading Machines III, IV •Timothy Leary's Mind Mirror •To Preserve, Protect and Defend •Tower Of Myraglen •Troll's 'MicroCoarseware' series •Webster: The Word Game •Word Munchers •Words At Work: Compound It •Words At Work: Suffix Sense •World Games •World's Greatest Baseball Game •World Karate Championship •Writer Rabbit •Zoyon Patrol ■ **APTs:** •Buck Rogers •Ikari Warrior •Kung-Fu Master •Leisure Suit Larry IIgs •Marble Madness •Realm Of Darkness •Rings Of Zilfin •Space Quest IIgs •Super Boulder Dash ■ **Playing Tips:** •2400 A.D. •Donkey Kong •Infiltrator •Space Quest IIgs •Spy Hunter •Swashbuckler •Thexder •Ultima II • ■ **Features:** •Modify Super IOB to read/write every other track •APT for Rings Of Zilfin: Turn yourself into a lean, mean fighting Machine •More Softkeys for M.E.C.C. software (1987) •How To Use The Electronic Art's RWTS •APT for Realms Of Darkness: Realm's Wrecker! •Putting Super Boulder Dash onto a hard disk **IBM** ■ **Softkeys:** •EasyWriter 1.0, II •Zork III

**52 February 1988** ■ **Features:** •The Product Monitor •Unprotecting The Unprotectable: Macintosh Softkeys! •A.P.T. Cornucopia •APT:Alternate Reality-Dungeon: Create A Super-human •Softkey for SSI's RDOS disks: 1.ProDOS RDOS, 2.RDOS Transfer Utility •Making Cracked II Plus Disks Work On The IIc ■ **Softkeys:** •Apple Gradebook v2.6 •Award Maker Plus •Black Cauldron •Black Magic •California Games •Car Builder •Color Print Shop •Computer Ambush •Concepts In Science •Disney's Comic Strip Maker •Elite •Empire I, II •European Nations & Locations •Fooblitsky •Grid Designer •H.E.R.O. •Ikari Warriors •Infiltrator II •Le Francais par Ordinateur •Little Computer People's House on a Disk ★ Main Street Filer •Master Diagnostics IIe ★ MegaFiler ★ MegaMerge •Microzine 23 •Might & Magic ★ Millionaire •Mindplay software •Music Construction Set •Nibbler •Operation Market Garden •Phantasie ★ Planetfall •PrintMaster Plus •Print Shop •Questron •Regatta •Ring Quest •Ringside Seat •Rings Of Zilfin •Shanghai •Silent Service •Snooper Troops •Spy's Adventure in N. America •Super Print •Tass Times In Tonetown •Think Quick ★ Transylvania •Ultima I re-release •Where in the USA is Carmen Sandiego •World

Games ★ Zork I (★ MacIntosh softkey) ■ **A.P.T.s:** Alternate Reality: The Dungeon •Arctic Fox •Bard's Tale II •Beyond Zork •Black Magic •Cavern Creatures •Drol •Goonies •Ikari Warriors •Zorro ■ **Playing Tips:** •Beyond Castle Wolfenstein •Championship Lode Runner •Conan •King's Queen II •Lode Runner •Lurking Horror •Station Fall •Ultima IV •Zork I ■ **Softkeys:** •Lotus 1-2-3 •Flight Simulator •PFS Report I ■ **A.P.T.S.:** •Bard's Tale

**51 January 1988** ■ **Features:** •The Crypt-arithmetic Helper •Using EDD IV to Modify Tracks And Sectors •Bard's Tale APT: Dungeon Mapper Revisited •RAMfactor mod for Laser I 28 •Ultima IV APT edit-tables •The Product Monitor •Get Better Sound by using the cassette jacks •Making A Fast Boot Disk •Might & Magic APT edit-tables ■ **Softkeys:** •2400 AD •Aliens •Alphabet Zoo •Amnesia •Bag Of Tricks •Bard's Tale I •Bard's Tale II •Battle Cruiser •Beach-head II •Below The Root •Black Magic •Body Awareness •Bridge 4.0 •Carriers At War •Catalyst 3.0 •Centipede •Championship Boxing •Championship Wrestling •Chessmaster 2000 •Combining The Elements •Commando •Creative Contraptions •Einstein Compiler •Fat City •Fight Night •Flight Simulator v2.0 •Fun with Direction •GBA 2-On-2 Championship Basketball •GraphicWriter v1.1 RA •Growing Up Small •House-on-a-disk •Intrigue •Jet •Jungle Hunt •Kindercomp •Knowing Numbers •Kung-fu Master •Law Of The West •Learning Well series •Letters And Words •Little Computer People •Make Your Own Murder Party •Manic Mansion •Master Diagnostics •Movie Maker •Music Construction Set •Pinball Construction Set •Pitstop •Print Shop Graphics Library Holiday •Print Shop IIgs •Rendezvous •Shapes And Patterns •Silent Service •Sorcerer •Spy vs Spy I & II •Stargate •Stellar 7 •Stickybear ABCs •Stickybear Drawing •Stickybear Numbers •Stickybear Printer •Stickybear Printer Library I & II •Stickybear Townbuilder •Super Boulderdash •Temple Of Apshai Trilogy •Tomahawk •Thexder •Walt Disney's Card And Party Shop •Walt Disney's Cartoon Maker •Wings Of Fury •Word Maze •World's Greatest Baseball Game •Zork III ■ **A.P.T.s:** Bard's Tale •Lode Runner •Might & Magic •Ultima IV •W. Disney's Card And Party Shop •Wizardry III •Wizardry IV ■ **Playing Tips:** •Autoduel •King's Quest •Manic Mansion •Summer Games •Tass Times In Tonetown •Thexder •Where In World is Carmen Sandiego?

**50 December 1987** ■ **Features:** •Super Boulderdash APT-writer •Softkeys to Activation/ MECC/ and PFS ProDOS/ software •Double F-8 ROM space w/o motherboard surgery •Ace-Apple bimodal Switch •Using Sider hard drives 3 1/2", 800K drives, & 5 1/4" drives in DOS 3.3 ■ **Softkeys:** •Aliens •Alter Ego •Alternate Reality •Amazing Reading Machines •Amazon •American Challenge •Arcade Album #1 •Arithmetic Critters •Award Maker •Baseball Database •Bard's Tale II: Destiny Knight



# COMPUTIST back issues

To order, Use the back issue order form on page 42

**50** continued: •BC's Quest for Tires •Bop & Wrestle •Champ. Boxing •Champ. Wrestling •Clock Works •Commando •Computer Prep for SAT •Conflict In Vietnam •Counting Critters •Crisis Mountain •Dataquest 50 States •Deluxe Paint II •Dino Eggs •Disney Card & Party Shop •Disney Comic Strip Maker •Draw Plus •Eidolon •Electric Crayon ABCs •Expedition Amazon •Facemaker •First Letter Fun •Fish Scales •Fun From A-Z •Game Maker •GBA Champ. Basketball •GFL Champ. Football •Graphicwriter 1.0R/1.1R •Great Road Race •Hacker II •Hardball •Infiltrator II •Instant Music •James Bond 007: A View To A Kill •Jenny's Journeys •Kung Fu Master •Little People •List Handler •Manic Mansion •Mastery Arithmetic Games •Market Place •Master of Lamp •Math Rabbit •Microzine #17 •Might and Magic •Mission In Solar System •Moebius •Music Construction Set •Music Studio •Number Munchers •Paint With Words •Paintworks Plus •Path Tactics •pfs:File •pfs:Graph •pfs:Plan •pfs:Report •pfs:Write •Phonics Prime Time •Portal •Principal's Assistant •Print Shop ProDOS 8 v1.4 •Print Shop Holiday Edition •Quickflash! •Reader Rabbit •Realm of Impossibility •Robot Odyssey 1 v.2.0 •Rocky Horror Show •Rocky's Boots v4.0 •Saracen •Shanghai •Silent Service •Skylab •Sound Tracks •Speedy Math •Spindizzy •Street Sports Baseball •Sub-Mission •Super Boulderdash •Tass Times in Tonetown •Thexder •Top Fuel Eliminator •Word Handler •Word Munchers •Words at Work •World Karate Champ. •Writer's Choice: Elite •Zardax v5.2.1

**49** November 1987 ■Features: •Eliminate some ProDOS erroneous error messages •Date/time without a clock card •Sector surgery: recover lost files •Generating Applesoft programs 'on-the-fly' •Product Monitor reviews ■PLUS: How to convert List Handler files into standard text files •How to make GRAPHIC.GRABBERv3 run on the ligs •Laser 128 'absolute' RESET ■Playing Tips: •Bard's Tale II •Conan •Donkey Kong •Hacker I •Hard Hat Mack •Orbitron •Print Shop Companion •Spellbreaker •Spy Hunter •Ultima 4 ■A.P.T.s: Infiltrator •Lode Runner •Montezuma's Revenge •Swordthrust series ■Softkeys: •Addition Logician •Animate •Arcade Boot Camp •Arctic Fox •Bard's Tale II •Cat'n Mouse •Counting Critters •Dam Busters † •Destroyer •Draw Plus v1.0 •Dr. Ruth's Comp. Game Of Good Sex •Echo 1.0 •E.D.D. 4 •Gamemaker •Hard Ball •Infiltrator •List Handler † •Locksmith 6.0 Fastcopy † •Magic Slate •Math Critters •Millionaire •Mind Mirror •One On One •Paintworks Plus v1.0 •Paintworks Plus v1.1 •PHM Pegasus •Portal •Quotient Quest •Reader Rabbit •Saunders' Chemistry CAI •Science Toolkit •Shanghai •Strip Poker † •Super Bunny •Super Sunday •Swordthrust series † •Term Paper Writer •Thief •Top Fuel Eliminator •Typing! † •Up-n-Down •Willy Byte •Writer's Choice Elite v1.0 •Writing A Character Sketch •Writing A Narrative

**48** October 1987 ■Features: •Dungeon Editor & Encounter Editor for Ultima III •APT for Shadowkeep •Softkey for Shadowkeep •Softkey for Apple Business Graphics ■Softkeys: •816 Paint GS •Amnesia •Arctic Fox •Award Maker Plus •Bard's Tale II •Betterworking Word Processor •Beyond Castle Wolfenstein •Black Magic •Bookends Extended •Bop & Wrestle •Chess 7.0 •Chessmaster 2000 •Deluxe Paint GS •Destroyer •Hacker II •Hacker II GS •Hardball •Infiltrator •Instant Music GS •J-Bird •Mabel's Mansion •Marble Madness •Mean 18 GS Golf •Megabots •Might & Magic •Miner 2049er II •Mouse Word •Music Construction Set GS •Music Studio GS •New Oregon Trail •Paintworks Plus 1.0 GS •Paintworks Plus 1.0 GS •Paul Whitehead Teaches Chess •PHM Pegasus •Poetry Express •Print Shop color version •Rambo: First Blood part II •Rocky Horror Show •Sargon III\* •Shanghai GS •Spindizzy •TelePorter •Temple Of Apschai trilogy •Top Draw GS •Transylvania •Ultima I •World's Greatest Baseball Game

**47** September 1987 ■Features: •Infocom-text Reader Enhancement •Color Ultimapper mod to Ultimapper IV •Towne Mapper utility for Ultima IV •Dungeon Mapper utility for Bard's Tale ■Hardware Corner: Interrupting Your Apple •Softkey for Charlie Brown's 1,2,3s ■Softkeys: •Guitar Wizard •Gemstone Warrior •Notable Phantom •Micro Wine Companion •Stickybear Printer •Note Card Maker •Starcross •Wishbringer •Dinosaur Dig •Dam Busters •Pirate Adventure •Infiltrator •MECC software •Banner Catch •Turtle Tracks •PFS File •Microzine #12, #13, #14 •Marble Madness •Writer Rabbit •Arcticfox •Age Of Adventure •Might And Magic •Space Station •Alternate Reality •Mindshadow •Gemstone Warrior •Strip Poker •Lucifer's Realm •Manuscript Manager •Bank Street Writer III •Kids On Keys •The Missing Ring •Graphic Solution •Empire I, II •Champ. Golf

**46** August 1987 ■Softkeys: •Advanced Microsystems Technology programs •Word Attack •Star Blazer •Science Toolkit •The Color Enhanced Print Shop •Video Vegas •The Handlers •K.C. Deals On Wheels •Law Of The West •Break The Bank Blackjack •Foundation Course In Spanish •OGRE •Puzzles And Posters ■Features: •The Shift Key/Lower Case Option For II+ •Amazing Computer Facts •Shape Magic utility ■Review: Multiscribe

**45** July 1987 ■Softkeys: •Mouse Calc •Sands of Egypt •Number Farm •Agent U.S.A. •Wavy Navy •Kindercomp •Flight Simulator Update •Raid over Moscow •Crime Stopper •Key Perfect 5. •The Final Conflict •Miss Mouse •Snoggle ■Features: •Write Protecting the Microsoft RAM Card •Keys to Success on the Franklin Ace •Modified F8 ROMs on the Apple III ■Core: •Owner's Review of Copy Master II

**44** June 1987 ■Softkeys: •Arcade Boot Camp •Goonies •Zorro •Coveted Mirror •Crimson Crown •Compubridge •Fleet System 3 •Microwave •Escape •Catalyst 3.0 •Number Farm •Alphabet Circus •Joe Theisman's Pro Football •Black Cauldron •Intern. Gran Prix ■Features: •Making DOSless Utilities •Pixit Printer Drivers ■Review: Z-RAM Memory Expansion Board •Reading the Joystick

**43** May 1987 ■Softkeys: •Graphics Expander •Information Master •Certificate Maker •Elite •Catalyst 2.0 and 3.0 •Murder On The Mississippi •Temple Of Apschai Trilogy •Troll Associates programs •Spell It •Regatta •Cdex Training programs •Think Fast ■Features: •How to Write-Protect your Slot Zero •Capturing Locksmith 6.0 Fast Copy •Revisiting DOS to ProDOS and Back ■Core: •Computer Eyes / 2: a Review ■APTs: •Sword of Kadash & Rescue Raiders •Ultimaker IV

**42** April 1987 ■Softkeys: •Light Simulator •Beach-Head •Monty Plays Scrabble •Racter •Winnie the Pooh •Infocom Stuff, Kabul Spy, Prisoner II •Wizardry I & 2 •Lucifer's Realm •The PFS Series •Dollars and Sense •Strip Poker •Coveted Mirror •Wizard's Crown •The Swordthrust Series •Axis Assassin •Manuscript Manager •The Crown of Arthain •Address Book •Decimals 3.0 •Dragonfire ■Features: •Auto Duel Editor •Wizard's Crown Editor •Questron Mapper ■Core: •The Games of 1986 in Review ■Adventure Tips: •Ultima IV

**41** March 1987 ■Softkeys: •The Periodic Table •Gemstone Warrior •Inferno •Frogger •Story Maker •Adventure Writer •Mummy's Curse •Zaxxon •The Quest •Pitfall II •H.E.R.O. ■Features: •A Two-Drive Patch for Winter Games •Customizing the Speed of a Duodisk •Roll the Presses Part Two: Printshop Printer Drivers •The Games of 1986

**40** February 1987 ■Softkeys: •Adventure Writer •E-Z Learner •Mychess II •Raster Blaster •Cranston Manor •Ghostbusters •Designer's Pencil •The American Challenge •Encyclopedia Britannica Programs •Crime Wave ■Features: •Taking the Wiz out of Wizardry •Adding a Printer Card Driver to Newsroom ■Core Games of 1986

**39** January 1987 ■Softkeys: •MIDI/8+ •Homeword v2.1 •Borrowed Time •Amazon •Speed Reader II •Discovery! •M-ss-ng L-nks series •Donald Ducks's Playground •Mastering the SAT •Copy II Plus 4.4C •Master of the Lamps •One on One •Bridge Baron •A.E. •Great American Cross-Country Road Race •Computer Preparation for the SAT •Castle Wolfenstein •Luscher Profile •Skyfox •Silent Service •Echo Plus •Swashbuckler •Randamn ■Features: •Electronic Disk Drive Swapper

# COMPUTIST back issues

To order, Use the back issue order form on page 42

**37** continued • Abusing the Epilogues  
• Print Shop Companion's Driver Game ■ *Core*  
• Keyboard Repair • Fixing the Applesoft Sample Disk

**38** December 1986 ■ *Softkeys*:  
• Cyclod • Alternate Realty • Boulder Dash I & II  
• Hard Hat Mack (Revisited) • The Other Side • F-15 Strike Eagle • Championship Lode Runner • Gato V 1.3 • I, Damiano • Wilderness • Golf's Best  
■ *Features*: • The Enhanced/ Unenhanced II  
• Looking into Flight Simulator's DOS ■ *Core*  
• Appavarex • Installing a RAM disk into DOS 3.3

**37** November 1986 ■ *Softkeys*:  
• Under Fire • Pegasus II • Take I (revisited) • Flight Simulator II v1.05 (part 2) • Magic Slate • Alter Ego • Rendezvous • Quicken • Story Tree • Assembly Language Tutor • Avalon Hill games • Dark Crystal ■ *Features*: • Playing Karateka on a IIc • Track Finder • Syk to Dif ■ *Core*: • Breaking In: tips for beginners • Copy II Plus 6.0: a review • The DOS Alterer

**36** October 1986 ■ *Softkeys*: • Flight Simulator II v 1.05 • AutoDuel • Critical Reading • Troll's Tale • Robot War • General Manager • Plasmania • Telarium Software • Kidwriter v1.0 • Color Me ■ *Features*: • ScreenWriter meets Flashcard • The Bus Monitor • Mousepaint for non-Apples ■ *Core*: • The Bard's Dressing Room ■ *APT*: • Championship Lode Runner

**35** September 1986 ■ *Softkeys*:  
• Olympic Decathlon • Hi-res Cribbage • Revisiting F-15 Strike Eagle • Masquerade • The Hobbit • Pooyan • The Perfect Score • Alice in Wonderland • The Money Manager • Good Thinking • Rescue Raiders ■ *Feature*: • Putting a New F8 on Your Language Card ■ *Core*: • Exploring ProDOS by installing a CPS Clock Driver

**34** August 1986 ■ *Softkeys*: • Crisis Mountain • Terripin Logo • Apple Logo II • Fishies 1.0 • SpellWorks • Gumball • Rescue at Rigel • Crazy Mazey • Conan • Perry Mason: The Case of the Mandarin Murder • Koronis Rift ■ *Feature*: • More ROM Running ■ *Core*: • Infocom Revealed

**33** July 1986 ■ *Softkeys*: • Word Juggler • Tink! Tonk! • Sundog v2.0 • G.I. Joe & Lucas Film's Eidolon • Summer Games II • Thief • Instant Pascal • World's Greatest Football Game • Graphic Adventure #1 • Sensible Grammar & Extended Bookends • Chipwits • Hardball • King's Quest II • The World's Greatest Baseball Game ■ *Feature*: • How to be the Sound Master ■ *Core*: • The Mapping of Ultima IV

**32** June 1986 ■ *Softkeys*: • Revisiting Music Construction Set • Cubit • Baudville Software • Hartley Software • Bridge • Early Games for Young Children • Tawala's Last Redoubt • Print Shop Companion • Cracking Vol II • Moebius • Mouse Budget, Mouse Word & Mouse Desk • Adventure

Construction Set ■ *Feature*: • Using Data Disks With Microzines ■ *Core*: • Super IOB v1.5 a Reprint

**31** May 1986 ■ *Softkeys*: • Trivia Fever • The Original Boston Computer Diet • Lifesaver • Synergistic Software • Blazing Paddles • Zardax • Time Zone • Tycoon • Earthly Delights • Jingle Disk • Crystal Caverns • Karate Champ ■ *Feature*: • A Little Help With The Bard's Tale ■ *Core*: • Black Box • Unrestricted Ampersand

**30** April 1986 *Softkeys*: • Millionaire • SSI's RDO5 • Fantavision • Spy vs. Spy • Dragonworld • King's Quest • Mastering the SAT • Easy as ABC • Space Shuttle • The Factory • Visidex 1.1E • Sherlock Holmes • The Bard's Tale • *Feature*: • Increasing Your Disk Capacity • *Core*: • Ultimaker IV, an Ultima IV Character Editor

**29** March 1986 *Softkeys*: • Threshold • Checkers v2.1 • Microtype • Gen. & Organic Chemistry Series • Uptown Trivia • Murder by the Dozen • Windham's Classics • Batter Up • Evelyn Wood's Dynamic Reader • Jenny of the Prairie • Learn About Sounds in Reading • Winter Games • *Feature*: • Customizing the Monitor by Adding 65C02 Disassembly • *Core*: • The Animator

**28** February 1986 *Softkeys*: • Ultima IV • Robot Odyssey • Rendezvous • Word Attack & Classmate • Three from Mindscape • Alphabetic Keyboarding • Hacker • Disk Director • Lode Runner • MIDI/4 • Algebra Series • Time is Money • Pitstop II • Adventure to Atlantis • *Feature*: • Capturing the Hidden Archon Editor • *Core*: • Fingerprint Plus: A Review • Beneath Beyond Castle Wolfenstein (part 2)

**27** January 1986 *Softkeys*: • Microzines 1-5 • Microzines 7-9 | Microzines (alternate method) • Phi Beta Filer • Sword of Kadash • Another Miner 2049er • Learning With Fuzzywomp • Bookends • Apple Logo II • Murder on the Zinderneuf • *Features*: • Daleks: Exploring Artificial Intelligence • Making 32K or 16K Slave Disks • *Core*: • The Games of 1985: part II

**26** *Softkeys*: • Cannonball Blitz • Instant Recall • Gessler Spanish Software • More Stickybears • Financial Cookbook • Super Zaxxon • Wizardry • Preschool Fun • Holy Grail • Inca • 128K Zaxxon • *Feature*: • ProEdit • *Core*: • Games of 1985 part I

**25** *Softkeys*: • DB Master 4.2 • Business Writer • Barron's Computer SAT • Take I • Bank Street Speller • Where In The World Is Carmen Sandiego • Bank Street Writer 128K • Word Challenge • Spy's Demise • Mind Prober • BC's Quest For Tires • Early Games • Homework Speller • *Feature*: • Adding IF THEN ELSE To Applesoft • *Core*: • DOS To ProDOS And Back

**24** *Softkeys*: • Electronic Arts software • Grolier software • Xyphus • F-15 Strike Eagle

• Injured Engine • Mr. Robot And His Robot Factory • Appicillin II • Alphabet Zoo • Fathoms 40 • Story Maker • Early Games Matchmaker • Robots Of Dawn • *Feature*: • Essential Data Duplicator copy parms • *Core*: • DOS-Direct Sector Access

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

**22** *Softkeys*: • Miner 2049er • Lode Runner • A2-PBI Pinball • The Heist • Old Ironsides • Grandma's House • In Search of the Most Amazing Thing • Morloc's Tower • Marauder • Sargon III • *Features*: • Customized Drive Speed Control • Super IOB version 1.5 • *Core*: • The Macro System

**20** *Softkeys*: • Sargon III • Wizardry: Proving Grounds of the Mad Overlord and Knight of Diamonds • The Report Card VI.1 • Kidwriter • *Feature*: • Apple II Boot ROM Disassembly • *Core*: • The Graphic Grabber v3.0 • Copy II+ 5.0: A Review • The Know-Drive: A Hardware Evaluation • An Improved BASIC/Binary Combo

**19** *Softkeys*: • Rendezvous With Rama • Peachtree's Back To Basics Accounting System • HSD Statistics Series • Arithmetickle • Arithmekicks and Early Games for Children • *Features*: • Double Your ROM Space • Towards a Better F8 ROM • The Nibbler: A Utility Program to Examine Raw Nibbles From Disk • *Core*: • The Games of 1984: In Review-part II

**16** *Softkeys*: • Sensible Speller for ProDOS • Sideways • Rescue Raiders • Sheila • Basic Building Blocks • Artsci Programs • Crossfire • *Feature*: • Secret Weapon: RAMcard • *Core*: • The Controller Writer • A Fix For The Beyond Castle Wolfenstein Softkey • The Lone Catalog Arranger Part I

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

If you are also interested  
in obtaining out-of-print back-issues

Please see the descriptions  
listed on page 41



# Out-of-print back issues

To order, Use the back issue order form on page 42

Due to popular demand, these sold-out issues are available now as 'zeroxed' copies, full-sized and center stapled.

**21** ■ *Softkeys*: •DB Master version 4+ •Dazzle Draw •Archon •Twerps ■ *Readers' Softkeys*: •Advanced Blackjack •Megaworks •Summer Games •College Entrance Exam Prep •Applewriter revisited ■ *Features*: •Demystifying The Quarter Track ■ *Core*: •Proshadow: A ProDOS Disk Monitor

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

**17** ■ *Softkeys*: •The Print Shop •Crossword Magic •The Standing Stones •Beer Run •Skyfox •and Random House Disks ■ *Features*: •A Tutorial For Disk Inspection and the Use Of Super IOB •S-C Macro Assembler Directives (reprint) ■ *Core*: •The Graphic Grabber For The Print Shop •The Lone Catalog Arranger Part Two

**15** ■ *Softkeys*: •Mastertype •Stickybear BOP •Tic Tac Show ■ *Reader's Softkeys*: •The Financial Cookbook •Escape from Rungistan •Alien Munchies •Millionaire •Plato ■ *Features*: •MREAD/MWRT Update ■ *Core*: •A Boot from Drive 2 •DB Master's Data Compression Techniques ■ *Whiz Kid*: •DOS and the Drive - Part One ■ *Adventure Tips*: •Time Zone •Mission Asteroid •Enchanter •Zork I •Ultima •Ultima II •Death in the Caribbean •Gruds in Space •Zork III •Starcross

**14** ■ *Features*: •Super IOB v1.2 Update •Putting Locksmith 5.0 Fast Copy Into a Normal Binary File •Batman Decoder Ring •A fix for DiskEdit ■ *Softkeys*: •Seadragon •Rocky's Boots •Knoware •PFS Software •Computer Preparation SAT •MatheMagic ■ *Review*: •Boulder Dash

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

**12** ■ *Softkeys*: •Zoom Graphix •Flip Out •Lion's Share •Music Construction Set ■ *Reader's Softkeys*: •Hi-Res Computer Golf II •Suicide •Sabatage •Millionaire •Time is Money •Type Attack ■ *Features*: Pseudo-ROMs on the Franklin Ace ■ *Core*: •Psychedelic Symphony •The CORE Disk Searcher •The Armonitor ■ *Adventure Tips*: •Cranston Manor •Enchanter •Kabul Spy •Colossal Caves •The Witness •Pirate Adventure •Ultima III-Exodus •Adventureland

**11** ■ *Softkeys*: •Sensible Speller •Exodus: Ultima III ■ *Readers' Softkeys*: •SoftPorn Adventure •The Einstein Compiler v5.3 •Mask of The Sun ■ *Features*: •Copy II Plus v4.4C: Update Of An Old Friend •Parameter List For Essential Data Duplicator ■ *Core*: •Ultimaker III •The Mapping of Ultima III •Ultima II...The Rest Of The Picture

**10** ■ *Softkeys*: •Arcade Machine •Bank Street Writer •Minit Man ■ *Reader's Softkeys*: •Senible Speller IV •EDD IV •\*Krell LOGO •Canyon Climber ■ *Features*: •The Controller Saver •Examining Protected Applesoft BASIC Programs •Crunchlist II ■ *Core*: •Applear - Voice Aynthesis •Introducing the 655C802 and 655C816 Chips •Review - Dino Eggs ■ *Adventure Tips*: •Cranston Manor •Zork I •Planetfall •Mission Asteroid •Time Zone •Suspended •Critical Mass •Zork II •Castle Wolfenstein

**9** ■ *Softkeys*: •Sensible Speller •Sierra-On-Line Software •The Visible Computer: 6502 ■ *Reader's Softkeys*: •Visidex •Music Construction Set •Gold Rush •Visiterm •Cosmic Combat ■ *Features*: •Super IOB ■ *Adventure Tips*: •Pirate Adventure •Mask of the Sun •Colossal Caves •Transylvania •Death in the Caribbean •Zork II ■ *Core*: •Word Search Generator •ProDOS to DOS •ProDOS on a Franklin Ace

**8** ■ *Softkeys*: •Robotron •Legacy of Llylgamyn •The Artist •Data Factory v5.0 •EDD IV ■ *Reader's Softkeys*: •Spy Strikes Back •Hayden Software •Apple LOGO ■ *Features*: •Review of the Bit Copiers ■ *Core*: •COREfiler •ProDOS Data Encryptor ■ *Adventure Tips*: •Ulysses and The Golden Fleece •Serpentine •Ultima II •Castle Wolfenstein •Death in the Caribbean •Zork I •Zork II •Gruds in Space •Enchanter •Infidel •Serpent's Star ■ *Whiz Kid*: •How Data is Stored on Disk

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

**6** ■ *Softkeys*: •Pandora's Box •Donkey Kong •Caverns of Freitag •Visifile ■ *Features*: •Program Enhancements: Quick.Bug •Personalizing A Program •Modified ROMs ■ *Review*: •Essential Data Duplicator •The CIA ■ *Core*: •Data Bases

**5** ■ *Softkeys*: •Homeword •Aztec •Bag of Tricks •Egbert II •Starcross •Hard Hat Mack •The Home Accountant ■ *Reader's Softkeys*: •Dark Crystal •Screenwriter II •Visifile •Lancaster •Bill Budge's Trilogy of Games •Sammy Lightfoot •Amper-Magic •Buzzard Bait ■ *Feature*: •Getting on the Right Track

**4** ■ *Features*: Ultima II Character Editor ■ *Softkeys*: •Ultima II •Witness •Prisoner II •Pest Patrol ■ *Adventure Tips*: •Ultima II & III ■ Copy II Plus Pams Update

**3** ■ *Softkeys*: •Bag of Tricks •Multiplan ■ *Readers' Softkeys*: •Visiplot /Visitrend •Sneakers •Wizardry ■ *Features*: •No More Bugs: The Sequel •Hidden Locations Revealed •Map Maker ■ *A.P.T.s* •Choplifter ■ *Adventure Tips*: •Cranston Manor •Strange Odyssey

**2** ■ *Softkeys*: •Magic Window II •Multiplan ■ *Features*: •Parameters for Locksmith 4.1 •Page Flipper •String Plotter •Three-D Wall Draw ■ *Core*: Checksums ■ *Input*: •Reviews of unprotected commercial software

To order, Use the back issue order form on page 42



Send these issues and/or library disks:

Issue	Mag \$4.75	Disk \$9.95	Both \$12.95
54	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
53	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
52	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
51	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
49	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
48	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
47	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
46	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
45	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
44	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
43	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
42	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
41	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
40	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
39	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
38	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
37	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
36	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
35	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
34	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
33	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
32	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
31	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
29	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
28	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
27	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
26	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
25	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
24	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
★ 23	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
22	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
■ 21	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
■ 18	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
■ 17	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
★ 16	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
■ 15	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
■ 14	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
■ 13	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
■ 12	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
■ 11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
■ 10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
■ 9	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
■ 8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
■ 7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Core 3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
■ 6	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
■ 5	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
■ 4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
■ 3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Core 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
■ 2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Core 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Computing 3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

**Best of Hardcore Computing**...     
Only the disk is available

**Core Special Combo \$10**.....     
(All three CORE magazines; no disks)

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.

★ Limited supply; first-come-first-serve basis.  
● Not available.  
○ Out-of-print; only 'Zeroxed' copies for sale.

# COMPUTIST

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

## Back Issues and Library Disk Rates

- US, Canada and Mexico back issue rate - \$4.75 each.
- All other Foreign back issue rate - \$8.75 each.
- US, Canada, Mexico library disk rate - \$9.95 each.
- All other Foreign library disk rate - \$11.94 each.
- "Both" disk and magazine rates for:  
US, Canada & Mexico - \$12.95 each combination.  
All other Foreign - \$18.95 each combination.

## What is a library disk?

A library disk is a diskette that contains programs that would normally have to be typed in 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 55. If you wish to purchase a library disk not listed on the left (under the DISK column), used the out-of-print back issues ad on page 42.

## 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.....

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

Address \_\_\_\_\_

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

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

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

Signature \_\_\_\_\_ CP55

• US funds drawn on US bank. • Most orders 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. • Send check/money order to:

COMPUTIST PO Box 110846-T Tacoma, WA 98411 (206) 474-5750



# the COMPUTIST shopper

Software Package	II e	II gs	Software Package	II e	II gs
816 Paint	\$45.00 <input type="checkbox"/>	\$45.00 <input type="checkbox"/>	The Print Shop	\$34.00 <input type="checkbox"/>	\$40.00 <input type="checkbox"/>
Airheart	\$24.00 <input type="checkbox"/>		Print Shop Companion	\$27.00 <input type="checkbox"/>	
Award Maker Plus	\$24.00 <input type="checkbox"/>		Print Shop Graphics Library:		
Bank Street Writer 64K	\$46.00 <input type="checkbox"/>		Disk One	\$16.50 <input type="checkbox"/>	
Bank Street Speller	\$46.00 <input type="checkbox"/>		Disk Two	\$16.50 <input type="checkbox"/>	
Bank Street Mailer 64K	\$46.00 <input type="checkbox"/>		Disk Three	\$16.50 <input type="checkbox"/>	
Bank Street Filer 64K	\$46.00 <input type="checkbox"/>		Pro-Byter	\$32.00 <input type="checkbox"/>	
Bank Street Writer Plus (128K)	\$46.00 <input type="checkbox"/>		Racter	\$30.00 <input type="checkbox"/>	
Bank Street Mailer 128K	\$46.00 <input type="checkbox"/>		Reader Rabbit	\$27.00 <input type="checkbox"/>	\$34.00 <input type="checkbox"/>
Bank Street Filer 128K	\$46.00 <input type="checkbox"/>		Rocky's Boots	\$34.00 <input type="checkbox"/>	
Beagle Graphics	\$35.00 <input type="checkbox"/>		Science Tool Kit: Master Module	\$46.00 <input type="checkbox"/>	
Beyond Zork	\$27.00 <input type="checkbox"/>		Shanghai	\$24.00 <input type="checkbox"/>	\$30.00 <input type="checkbox"/>
Black Cauldron	\$27.00 <input type="checkbox"/>		Silent Service	\$23.00 <input type="checkbox"/>	
Captain Goodnight	\$24.00 <input type="checkbox"/>		Solo Flight	\$23.00 <input type="checkbox"/>	
Championship Karate	\$14.00 <input type="checkbox"/>		Space Quest	\$32.00 <input type="checkbox"/>	\$32.00 <input type="checkbox"/>
Clip Art Gallery (for Paintworks Plus)	\$20.00 <input type="checkbox"/>		Star Trek- The Kobayashi Alternative	\$27.00 <input type="checkbox"/>	
Copy II Plus v8.0	\$23.00 <input type="checkbox"/>	\$23.00 <input type="checkbox"/>	Star Trek II- The Promethean Prophecy	\$27.00 <input type="checkbox"/>	
Dazzle Draw	\$40.00 <input type="checkbox"/>		Stickybear Series:		
Destroyer	\$25.00 <input type="checkbox"/>		ABC's	\$24.00 <input type="checkbox"/>	
DiskQuirk	\$18.00 <input type="checkbox"/>		Math I	\$24.00 <input type="checkbox"/>	
Draw Plus	\$52.00 <input type="checkbox"/>		Math II	\$24.00 <input type="checkbox"/>	
F-15 Strike Eagle	\$24.00 <input type="checkbox"/>		Numbers	\$24.00 <input type="checkbox"/>	
Fantavision	\$34.00 <input type="checkbox"/>	\$40.00 <input type="checkbox"/>	Reading	\$24.00 <input type="checkbox"/>	
Flight Simulator II	\$36.00 <input type="checkbox"/>		Shapes	\$24.00 <input type="checkbox"/>	
Gamemaker	\$34.00 <input type="checkbox"/>		Typing	\$24.00 <input type="checkbox"/>	
The Graphics Studio	\$42.00 <input type="checkbox"/>		Writer	\$24.00 <input type="checkbox"/>	
Hacker II	\$25.00 <input type="checkbox"/>	\$30.00 <input type="checkbox"/>	Sub Mission	\$27.00 <input type="checkbox"/>	
Hardball	\$22.00 <input type="checkbox"/>	\$28.00 <input type="checkbox"/>	Summer Games	\$14.00 <input type="checkbox"/>	
Hitchhiker's Guide	\$20.00 <input type="checkbox"/>		Super Macroworks	\$30.00 <input type="checkbox"/>	
Infiltrator II	\$20.00 <input type="checkbox"/>		Tass Times in Tonetown	\$24.00 <input type="checkbox"/>	\$24.00 <input type="checkbox"/>
Jet	\$29.00 <input type="checkbox"/>		The Tower of Myraglen	\$32.00 <input type="checkbox"/>	
Karateka	\$24.00 <input type="checkbox"/>		Thexder	\$23.00 <input type="checkbox"/>	\$23.00 <input type="checkbox"/>
Knight of Diamonds	\$23.00 <input type="checkbox"/>		Topdraw	\$66.00 <input type="checkbox"/>	
Labrinth	\$20.00 <input type="checkbox"/>		Type	\$30.00 <input type="checkbox"/>	
Legacy of Lylgamyn	\$26.00 <input type="checkbox"/>		Typing Tutor IV	\$32.00 <input type="checkbox"/>	\$32.00 <input type="checkbox"/>
Loderunner	\$24.00 <input type="checkbox"/>		Video Vegas	\$18.00 <input type="checkbox"/>	
Macroworks	\$20.00 <input type="checkbox"/>		Visualizer	\$53.00 <input type="checkbox"/>	\$59.00 <input type="checkbox"/>
Math Blaster	\$28.00 <input type="checkbox"/>		Where in USA is Carmen San Diego	\$30.00 <input type="checkbox"/>	
Math Rabbit	\$27.00 <input type="checkbox"/>		Where in World is Carmen San Diego	\$30.00 <input type="checkbox"/>	
Mean 18, Ultimate Golf	\$28.00 <input type="checkbox"/>		Winter Games	\$25.00 <input type="checkbox"/>	\$27.00 <input type="checkbox"/>
Merlin ProDOS	\$66.00 <input type="checkbox"/>		Wizards	\$32.00 <input type="checkbox"/>	
Might & Magic	\$32.00 <input type="checkbox"/>		Word Attack	\$28.00 <input type="checkbox"/>	
Millionaire II	\$40.00 <input type="checkbox"/>		Wordperfect w/ Spelling Checker	\$95.00 <input type="checkbox"/>	\$95.00 <input type="checkbox"/>
Mousewrite	\$99.00 <input type="checkbox"/>	\$99.00 <input type="checkbox"/>	World Games	\$25.00 <input type="checkbox"/>	\$27.00 <input type="checkbox"/>
Multiscribe	\$45.00 <input type="checkbox"/>	\$66.00 <input type="checkbox"/>	Writer's Choice Elite	\$60.00 <input type="checkbox"/>	
Music Studio	\$52.00 <input type="checkbox"/>		Writer Rabbit	\$24.00 <input type="checkbox"/>	
Paintworks Plus	\$52.00 <input type="checkbox"/>		Zork Trilogy	\$46.00 <input type="checkbox"/>	
Portal	\$30.00 <input type="checkbox"/>				

**COMPUTIST**  
is proud to  
offer these  
**LOW software**  
prices\* for  
**Apple II and**  
IIgs software



\* Prices subject to  
change without notice.

Because we do not 'stock' but  
order only the software needed,  
the COMPUTIST Shopper keeps  
operating costs to a minimum  
and can pass the savings on to  
you.

In most cases, your order is  
shipped within two weeks. The  
COMPUTIST Shopper will NOT  
cash your check nor charge  
your credit card until your  
software is ready to be shipped.  
If the software you ordered is  
not available, you will have the  
option to cancel your order, or  
make an alternative selection.

### How To Order

- **US orders:** Check the box for your selection. For Apple IIgs software, check the box in the right-hand column.
- Please add \$3 per order for shipping & handling. Orders over \$200 receive **free** shipping.
- Most orders shipped UPS, so use your **street address**.
- **Washington state**, please add 7.8% sales tax.
- Offer good while supplies last.
- **Foreign Orders:** Please inquire as to appropriate shipping fees

Name \_\_\_\_\_ ID# \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_  
Country \_\_\_\_\_ Phone \_\_\_\_\_  
  \_\_\_\_\_ Exp. \_\_\_\_\_  
Signature \_\_\_\_\_ CP55

● Software  
SUBTOTAL \_\_\_\_\_  
● Plus  
SHIPPING \_\_\_\_\_  
(see above)  
● WA residents  
Sales Tax \_\_\_\_\_  
7.8 %  
■ TOTAL  
Enclosed \_\_\_\_\_

SoftKey Publishing PO Box 110816-T Tacoma WA 98411

## New IBM COMPUTIST?

In order to make IBM RDEX easier to read and understand, the following formats will be used.

**Commands you type are bold. Long lines tend to wrap down like this**

**But** *don't type these comments*

Remember, press **ENTER** after every command line. If something important is shown on the monitor after a command,

it will be shown like this

**DEBUG** is usually invoked in softkeys. It has several powerful single-letter commands listed below:

Assemble *address*  
 Compare *range/address*  
 Dump *range/address*  
 Enter *address list*  
 Fill *range list*  
 Go *address*  
 Hexarithmic *value value*  
 Input from *port-address*  
 Load file *address drive sector sector...*  
 Move memory *range address*  
 Name of file *drive path filename.ext*  
 Output control *port-address*  
 Proceed *address value*  
 Quit debug  
 Register *register-name*  
 Search *range list*  
 Trace *address value*  
 Unassemble *address/range*  
 Write *address drive sector sector...*

Mike Basford

Some of these are from BBS...

## Flight Simulator:: RGB Modifications

### ■ Requirements

- At least 96K

The modifications included herein will allow an RGB monitor to show some colours using the *Flight Simulator* program. This modification is not perfect, nor is it very well tested.

The user should make a copy of his *Flight Simulator* program using his favorite technique (see COMPUTIST #52)

I did find a bug: when the user enters the slow mode, these modifications are nullified.

Basically, the technique is to intercept the disk vector and setup a port for the colour display adapter for the needed values. I

certainly hope, that by disclosing this technique, Microsoft doesn't skin my hide.

Anyway, to modify your extra spare disk, boot up *DEBUG* in DOS 2.0 and type the following:

*Note: you need a system with at least 96K to use this modification as is.*

**l cs:0 0 0 1** *Flight Simulator disk in drive A.*  
**a 0**

```
mov ax,201
mov dx,0
mov CL,2
mov ch,27
mov bx,1000
mov es,bx
xor bx,bx
int 13
jmp 1000:0
```

**w cs:0 0 0 1**  
**l cs:0 0 139 1** *Flight Simulator Disk in Drive A.*  
**a 0**

```
push cs
pop ds
mov ax,0
mov es,ax
es:
mov ax,[4c]
mov [70],ax
es:
mov ax,[4e]
mov [72],ax
mov ax,48
es
mov [4c],ax
es
mov [4e],cs
cli
xor ax,ax
mov ds,ax
mov es,ax
mov ss,ax
mov sp,c0b0
mov cx,200
mov SI,7c00
mov DI,500
sti
repz movsb
jmp 0:7c18
```

**a 48**

```
pushf
push cs
CS:
mov [74],ax
mov ax,5b
push ax
cs:
mov ax,[74]
jmp F000:EC59 This is also saved at 1000:0070
pushf
```

```
push ax
push dx
mov dx,3d8
mov AL,0a
out dx,AL
inc dx
mov AL,20 see note below
out dx,AL
pop dx
pop ax
popf
iret
```

**W CS:0 0 139 1**

*Note: The value 20 a couple of lines up sets up the colours for low intensity cyan/magenta/white. Good luck, and may the colors be with you!*

*Softkey for...*

## Samna Word II v1.1

?

**1** Copy all the files from SAMNA DISK 1 to a formatted work disk or hard disk volume. All files should be copied into the same directory if using DOS 2xx.

**2** Copy all the files from SAMNA DISK 2 to the same work disk or hard disk volume as in Step 1.

Now, you will need to "patch" three programs on the working disk using *DEBUG*.

**3** Log onto the drive that contains the working disk (e.g. b:)

**b:**

**4** Have *DEBUG* on a disk or volume somewhere.

**5** First, we'll patch *SAMNA.COM*. Each line below should be typed exactly as shown with the **ENTER** key after it.

```
debug samna.com
e 5f1 90 90
e 14be 90 90
w
q
```

**6** Next, patch *FILE2*

```
debug file2
e c99 e9 43 5f
w
q
```

**7** Finally, patch *INSTALL.COM*

```
debug install.com
e 1feb 90 90
w
q
```

There, we're finished with the patches. The patch to *INSTALL.COM* is only needed to install your type of printer



## Installing the Printer

**1** Log on to the drive containing your patched working SAMNA disk.

**2** Place the SAMNA PRINTER DISK (a copy or write-protected original) into another disk drive.

**3** Type:

### INSTALL

**4** When it asks if you are installing or making a change, press **C**.

**5** When it asks for the drive that has the original SAMNA PRINTER DISK, reply with the letter of the drive that has the SAMNA PRINTER DISK.

**6** When it asks for the drive that you are installing on, enter the drive letter of the working SAMNA DISK.

**7** Then you will be given a screen of printer choices. Choose the one you want (refer to the SAMNA manual), and press **ENTER**.

**8** That completes the printer installation. (Files *DEFAULT.PAR* and *SAMNA.OV7* are changed. The printer file is copied from the printer disk and renamed to *SAM NA.OV7* and the file *DEFAULT.PAR* is changed)

That completes the unprotection.

Remember, all files should be in the same directory. This is different from the way SAMNA is set up to work normally. (IE. The file *SAMNA.COM* is normally in the root directory and the other files are in a sub-directory called *SAMNA*.) This works OK on floppies and hard disks but not on some networks.

I have found SAMNA WORD II to be a nice product but it's usefulness is diminished by it's installation difficulty and copy-protection method.

This will make hard-disk usage easier and backup/restore operations more reliable.

IBM RDEX END

## Help Wanted

The IBM RDEX editor desperately needs part-time writers. Any IBM experience will do. If you've used your IBM in some unusual way or solved some problems or just found something neat to do, write to the IBM RDEX editor and let him know.

Tell your friends about the IBM RDEX. Put messages on all the bulletin boards you frequent.

Get the word out!

Send all contributions to:

IBM-RDEX editor  
COMPUTIST  
PO Box 110846-T  
Tacoma, WA 98411

## REACH FOR THE POWER TRIO.



- Create programs with super high resolution graphics in up to 256 colors at once, plot shapes and text fonts in color, produce animated effects and more, with **ICONIX**.
- Sample and record your own sounds, compose music in up to 30 voices, create sound effects and much more, with **SONIX**.
- Trace and locate files, recover deleted files, create a mouse-driven program launcher and much more, with **DISC COMMANDER**.

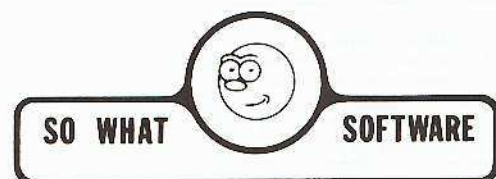
With "The Power Trio" from So What Software, now you can create your own BASIC software programs using the full range of graphics and sound power of the Apple IIG's.

Create colorful animated programs using the super high resolution power of **ICONIX**. Orchestrate compositions through the full sound and music environment of **SONIX**. Or trace and locate files with the powerful disk editor, **DISC COMMANDER**, and much more.

Easy-to-use and very affordable, "The Power Trio" will turn your BASIC programs into FANTASTIC programs!

**SONIX, ICONIX, and DISC COMMANDER:** the hands-on favorites!

Call us today, at (714) 964-4298



10221 SLATER AVENUE, SUITE 103, FOUNTAIN VALLEY, CA 92708



# CLASSIFIED ADS

## FOR SALE: Software

**A HACKER'S CHALLENGE** is for hackers of the APPLE II. In it are graphical game-modules you must play to discover **CLUES**. Your skills of DOS 3.3 lore & utility usage helps you to **FIND** and "cheat" your way into the games since they're hidden all over the disk. Your goal is to solve its **GAME-SECRET** that could win you an APPLE II game prize if you'r one of the 1st ten to solve it! This two sided 5.25" game comes with a DOS 3.3 and hint booklet. Send \$9.95 to Mark V. Whitehurst, Box 485, Franklin Park, IL 60131. Game demo disk only \$2.

**RUINS of KELEDOOR** - One of the 1st 3-D dungeon games with detailed and colored walls, monsters, pit-traps, secret doors, teleports, battle and movement sound effects to enhance game play. Your position is saved to the disk at any time. This two-sided 5.25" APPLE II game is not protected and comes with an adventure booklet. Send check or M.O. for \$12.95 to Mark Whitehurst, Box 485, Franklin Park, IL 60131. \$2 for demo disk. Dungeon maps only \$1.

**TRIVIA PYRAMID** is a hi-res game of 1,000 questions you and up to three friends can play - or just you against the computer. You try to get your marker to the top of the pyramid gameboard by answering trivia correctly. Options include the use of a timer, 2-4 players (one can be the computer), selecting which of the eight trivia groups to use in the game. An included "Trivia Maker" utility allows you to create Trivia Disks of your own! This two sided 5.25" APPLE II game is not protected and comes with a booklet. Send \$12.95 to Mark Whitehurst, Box 485, Franklin Park, IL 60131. Demo disk is only \$2.

Mind Prober—\$9, Practicalc—\$9  
Jingle Disk—\$1, Cardware—\$2  
FFS.Write, File, Report—\$45  
Postpaid. Byron Blystone, P.O.  
Box 1313, Snohomish, WA 98290.

### BIBLEWORKS

The text of the entire King James version in Appleworks W/P files. Set of seven 3.5" disks — \$69.95  
Texas residents add 7% sales tax  
LTE, Box 777, Refugio TX 78377

### Index to COMPUTIST #1-53

Appleworks data base, contains all articles, softkeys, tips, bugs, & programs. Program, Publisher, Vol, Pg, over 1800 entries. Comes broken into small databases and as ASCII text files. Two double-sided disks.— \$6  
John R. Pierce, P.O. Box 111  
Hume, IL 61932.

### Softkeys: Appleworks database

Complete from start of COMPUTIST to #53. Includes **BUGS**, alphabetic & Master & Issue databases. Send \$4 to T.H. Carlock, P.O. Box 280525  
Lakewood, CO 80228

## FOR SALE: Hardware

### APPLE SALE!!

Scribe Printer \$149

Modem 300 \$159

80 column monitor \$79

Like new with cables  
and manuals (407)391-8424

## WANTED: Software

Trade your unwanted software. Send your list of programs to trade. I have over 70 originals to trade. Byron Blystone, P.O. Box 1313, Snohomish, WA 98290.

WANTED: "Apple Invaders" game by Programma, long out of business. Jeff Wagner, 566 Main St., Vermilion, OH 44089

## ESC

COMPUTIST BACK ISSUES. All issues except #25. \$200.00.  
C. Davis, 21926 Linn Ridge,  
Fairhope, AL 36532

## How to place an UnClassified Ad

For short ads, use this order form. For larger ads, submit typed copy with appropriate instructions. Remember, only 30 characters per line. \$5 for the first line, \$1 per line after that.

----- \$5  
----- \$6  
----- \$7  
----- \$8  
----- \$9  
----- \$10

**Special Graphics Instructions:** The first three words of the first line are printed in **bold** for free. If you want other lines bolded, write **BOLD==** in the first 6 spaces of that line. There is a \$1 charge for each line you want bolded. If you want a line centered, write **CENTER** on the first 6 spaces of that line. There is no charge for centering any line. If you want a line both Bold and Centered, write **CENTERBOLD** on the first 10 character spaces.

Name \_\_\_\_\_ ID# \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_  
Country \_\_\_\_\_ Phone \_\_\_\_\_  
Signature \_\_\_\_\_ Exp. \_\_\_\_\_  
CP53

Total Number of lines: \_\_\_\_\_ \$ \_\_\_\_\_00

Number of bold lines \_\_\_\_\_ \$ \_\_\_\_\_00

\$Total for a single ad \$ \_\_\_\_\_00 \*

Run my ad \_\_\_\_\_ number of times. Each run that I pay for now will cost me only 1/2 the total initial cost \$ \_\_\_\_\_00

Final Total \$ \_\_\_\_\_00

\*Washington orders add 7.8% sales tax. Send this form and a check-money order (funds drawn on US bank only) for the entire amount to:

COMPUTIST unCLASSIFIEDS PO Box 110846 Tacoma, WA 98411

# NEW!

## APPLE COMPATIBLE PRODUCTS

## LOW FACTORY DIRECT PRICES!

DEALERS CALL!

UNIV & SCHOOL P.O.'s WELCOME!

### ONE MEG RAM BOARDS? WE'VE GOT EM!

#### THE Super Expanders!

S.E. ligs	OK Now	\$49
S.E. lle	* OK Now	\$59
S.E. llc	* OK 1/88	\$69
S.E. ll+	* OK 3/88	\$79

Above with 256K =  
Above with 512K = **CALL**  
Above with 1MEG =

\* FREE Appleworks RAM  
Expansion Software!

### QUIET COOLING FANS!

ligs No Audio Noise	\$25
ll+/e w/Surge Protect	\$27

### OTHER PERIPHERALS!

64K/80 Column Bd	lle	\$35
Super 64K/80 Col	lle	\$49
16K RAM Board	ll+	\$35
128K RAM Board	ll+	\$65
80 Column Board	ll+	\$49
Super Serial Bd	ll+/e	\$49
Z80 CP/M Board	ll+/e	\$35
Numeric Keypad	lle	\$35
Graphic Par Bd w/Cble		\$45
Joystick	\$15 &	\$25

### A/B SWITCHBOX!

Centronics or Serial \$29

**ADD ON DISK DRIVE: 5 1/4**  
Specify Your Model \$119

**TWO YEAR WARRANTY ON ALL PRODUCTS**  
**CALL OR WRITE FOR ILLUSTRATED LISTINGS!**  
ADD \$3 Shipping / COD ADD \$2 / VISA/MC OKAY 0%

### NEXO DISTRIBUTION

914 E. 8TH ST., Suite 109, National City, CA 92050  
(619) 474-3328 10AM-6PM Mon-Fri



# The Hacker's Ultimate Copy & Deprotection Utility Super IOB Collection !

ALL of our Super IOB controllers (through 1986) in 'ONE' package!

COMPUTIST developed the ultimate copy program to remove copy protection from software:

### The Super IOB program.

Since the introduction of Super IOB, COMPUTIST has used this flexible program to deprotect (or partially deprotect) dozens of commercial programs with far ranging protection schemes.

Super IOB deprotects disks by using a modified RWTS (the subroutine in DOS which is responsible for the reading and writing of disk sectors) for reading from the protected disk and then using a normal RWTS for writing to the deprotected disk.

### This package contains:

- ▶ TWO DISKS (supplied in DOS 3.3). Each disk contains at least 60 Super IOB Controllers including the standard, swap, newswap and fast controllers. Also included is **version 1.5 of Super IOB**, the Csave program from COMPUTIST No. 13, and a Menu Hello Program that lists the available controllers and, when you select one, automatically installs it in Super IOB and RUNs the resulting program.\*
- ▶ A reprint of **Disk Inspection and the Use of Super IOB**, from COMPUTIST No. 17. This article explains how to write your own Super IOB controllers.
- ▶ **COMPUTIST No. 32**, which contains an extensive article detailing the hows and whys of Super IOB v1.5 and at least 5 articles using the new Super IOB program.
  - Several of the controllers deprotect the software completely with no further steps. This means that some programs are only minutes away from deprotection (with virtually no typing).
  - The issue of COMPUTIST in which each controller appeared is indicated in case further steps are required to deprotect a particular program.\*\*

### The SUPER IOB Collection

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

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

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

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



### ■ Yes, please send me The Super IOB Collection

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

US/Canada/Mexico for \$16.00

Other Foreign for \$20.00

Send to: **Super IOB Collection**  
PO Box 110846-T Tacoma, WA 98411

or call: (206) 474-5750 for  or  orders.

\*Requires at least 64K of memory.

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

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

Address \_\_\_\_\_

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

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

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

Signature \_\_\_\_\_ CP55

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



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

# The Book(s) of Softkeys

## Volume I Compiled from issues 1-5

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

## Volume II Compiled from issues 6-10

contains softkeys for: •Apple Cider Spider •Apple Logo •Artist Arcade Machine •Bank Street Writer •Cannonball Blitz •Canyon Climber •Caverns of Freitag •Crush, Crumble & Chomp •Data Factory V •DB Master •The Dic\*tion\*ary •Essential Data Duplicator I & III •Gold Rush •Krell Logo •Legacy of Llylgamyn •Mask Of The Sun •Minit Man •Mouskattack •Music Construction Set •Oil's Well •Pandora's Box •Robotron •Sammy Lightfoot •Screenwriter II v2.2 •Sensible Speller 4,4c,4.1c •Spy Strikes Back •Time Zone v1.1 •Visible Computer: 6502 •Visidex •Visiterm •Zaxxon •software for: •Hayden •Sierra Online  PLUS the ultimate cracking program: Super IOB 1.5 •••and more!

## Volume III Compiled from issues 11-15

contains softkeys for: •Alien Addition •Alien Munchies •Alligator Mix •Comp. Prep. SAT •Cut & Paste •Demolition Division. •DLM software •EA (Electronic Arts) software •Einstein Compiler 5.3 •Escape From Rungistan •Financial Cookbook •Flip Out •Hi-res Computer Golf II •Knoware •Laf Pak •Last Gladiator •Learning With Leeper •Lion's Share •Master Type 1.7 •MatheMagic •Minus Mission •Millionaire •Music Construction Set •One-on-one •Penguin software •PFS software •The Quest •Rocky's Boots •Sabotage •Seadragon •Sensible Speller 4 •Snooper Troops II •SoftPorn Adventure •Stickybear series •Suicide •TellStar •Tic Tac Show •Time Is Money •Transylvania •Type Attack •Ultima III Exodus •Zoom Graphics •Breaking Locksmith 5.0 Fast Copy  PLUS feature articles on • Csave • The Core Disk Searcher • Modified ROMs.

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

Address \_\_\_\_\_

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

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

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

Signature \_\_\_\_\_ CP55

Volume I - \$7.95  
+\$2 shipping/handling

Volume II - \$12.95  
+\$2 shipping/handling

Volume III - \$17.95  
+\$2 shipping/handling

All 3 volumes! - \$30.00  
+\$2 shipping/handling

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