

80 COMPUTIST

Canada & Mexico \$7

U.S. \$3.75

Table of Contents

Editorial Note	3	Jeff Hurlburt	21	Dinosaur Days v1.0	20
RDEX Contributors	23	Bob Igo	21	Empire	8
Most Wanted Softkeys	23	Blain Johnson	21	Fahrenheit 451	19
unClassifieds	23	Gary Wills	21	Fay's Word Rally	20
				GATO v1.3	21
				Greeting Card Maker	20
				Hostage	15
				Keef The Thief	8
				Magic Spells v2.0	19
				Magic Spells v2.1	19
				Mickey's Crossword Puzzle Maker	16
				Monsters and Make Believe v1.1 ...	20
				Pipe Dream	8
				Pipe Dreams	15
				Rear Guard	17
				Rendezvous with Rama	19
				Same or Different	20
				Teacher's Tool Kit	19
				Teacher's Tool Kit (Iic)	20
				War of the Lance	10
				Where in the USA is Carmen	
				San Diego?	9
				Windwalker GS	15
				Windwalker Iie	15
				APTs:	
				Space Rogue	16
				Wizardry III. See Keeping the Wiz in	
				Wiz III	
				Playing Tips:	
				Countdown	6
				Space Rogue	16
				IBM Softkeys:	
				Serve and Volley	22
				Welltris	22
				Softkeys:	
				Ancient Art of War	9
				Battle Chess	10
				Bridge 6.0	10
				Captain Blood GS	10
				Answers:	
				Edison	21
				Groucho	21
				Ross A. Holmes	21
				E.N. Hondrick	21
				Questions:	
				¿	8, 16, 17, 21, 22
				Features, Notes & such:	
				Add Copy II Plus file handling to	
				your BASIC progr. See A Copy II	
				Plus PLUS!	7
				Comments on the Beginner's Book ..	7
				Formatting 720K disks as 1.44M HD.	
				See Holey Megabytes!	6
				How to SAVE hexdumps as CDA's ..	8
				Logging ProDOS Drives	6
				The Basics of Kracking (part 7)	17
				The Basics of Kracking (part 8)	18
				Bitkeys:	
				Black Magic	16
				Guild of Thieves	16
				Gunslinger	16
				King's Quest Series	16
				Leisure Suit Larry	16
				Man Hunter: New York	16
				Police Quest	16
				Realms of Darkness	16
				Saracen	16
				Sierra Boot Disks	16
				Silicon Dreams	16
				Space Quest Series	16
				Ultima V	16
				Wizardry Series	16
				Xyphus	16

**Subscribe to
COMPUTIST**
We give you
More!

- I am:
- Renewing my current subscription
 - Changing my address (Please include last label)
 - A new subscriber.

Subscription rates:

- \$26 For new subscribers (and late renewals that missed 1 issue. U.S. regular/3rd Class Mail) Includes an extra \$2 for postage and handling. We will send your Starter Kit and 1st issue by 1st Class mail.
- \$24 U.S. regular renewal (3rd Class mail)
- \$34 1st Class (U.S. / Canada / Mexico)
- \$54 All other Foreign (Air mail)
- \$68 COMBO (1st Class plus Disk)
- \$95 Foreign COMBO (Air mail plus Disk)

VISA

Charge It!

MC

Name _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

Vis/MC _____ Exp. _____

Signature _____ #80

• Send US funds drawn on US bank • For regular subscriptions, please allow 4-8 weeks for 1st issue or add \$2 for postage and we will send your 1st issue by 1st Class mail. • Send check/money order to:

COMPUTIST 33821 E Orville Rd Eatonville WA 98328-9590 (206) 832-3055

COMPUTIST (80)
PO Box 242
Kapowsin WA 98344-0242
ADDRESS CORRECTION REQUESTED

BULK RATE
U.S. Postage
PAID
Kapowsin WA
Permit No. 6

COMPUTIST

Charles R. Haight
Jeff Hurlburt
Dave Goforth

Editor
Reviews
BBS

COMPUTIST is published by SoftKey Publishing. Address all inquiries to:

COMPUTIST
33821 East Orville Road
Eatonville, WA 98328-9590
(206) 832-3055

• 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 1990 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 SoftKey Publishing assumes no liability or responsibility for the products advertised in this newsletter. Although we are usually pretty much in agreement, any opinions expressed by the authors are not necessarily those of COMPUTIST magazine or SoftKey Publishing.

SUBSCRIPTIONS: Rates (for 8 issues):

U.S. \$24 Canada/Mexico ..\$34
U.S. 1st Class ..\$34 Other Foreign\$54

• Subscriptions are sold by number of issues and not by month or year. An 8 issue subscription means that you will receive 8 issues before you need to renew. It's when you will receive each issue that we're a little erratic about.

• Domestic Dealer rates: Call (206) 832-3055 for more information.

• Change Of Address: Let the U.S. Postal Service know that you are moving. Tell them that you want your mail forwarded. If your issue does not come to you in an envelope then you have a regular subscription and you must tell the USPS to forward your third class mail. Notify us as soon as you know your new address. When we receive your notice of change of address, we will send you an acknowledgement card. If you do not receive the acknowledgement card after 2 weeks, send another notice or call us direct.

Issues missed due to non-receipt of change of address may be acquired at the regular back issue rate.

We are not responsible for missing issues 90 days after mailing date. If you do not receive an issue at the usual time each month, please call or write.

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

Readers Data EXchange

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 the DOS 3.3 System Master disk).

Commands and control keys

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 preceded by "ctrl". An example of both is:

6 ctrl P

Type 6. Next, place one finger on the ctrl key and then press P. Don't forget to press the return key.

Other special combination keypresses include ctrl reset and open-apple ctrl reset. In the former, press and hold down the ctrl key then press the reset key. In the latter, press and hold down both ctrl and open-apple then press reset.

Software recommendations

The Starter Kit contains most of the programs that you need to "Get started". In addition, we recommend that you acquire the following:

- Applesoft program editor such as "Global Program Line Editor (GPLE)".
- Assembler such as "Merlin/Big Mac".
- Bit-copy program such as "Copy II Plus", "Locksmith" or "Essential Data Duplicator".
- Word-processor (such as AppleWorks).
- "COPYA", "FID" and "MUFFIN" from the DOS 3.3 System Master disk.

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.

Laser 128: Your ROM includes a forced jump to the monitor. Press ctrl return reset.

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 that changes the open-apple ctrl reset vector to point to the monitor. (This will void an Apple //c warranty since you must open the case to install it.)

Apple //gs: If you have the 2.x ROM, 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 (CALL -151) before running any protected programs and press # return. This will turn on two hidden CDAs, Memory Peeker and Visit Monitor. Thereafter press open-apple ctrl esc to go to the Desk Accessories menu. Select Visit Monitor and there you are. Use ctrl Y to exit.

Recommended literature

- Apple II Reference Manual (or IIe, IIc, etc.)
- DOS 3.3 & ProDOS manual
- Beneath Apple DOS & Beneath Apple ProDOS, by Don Worth and Pieter Lechner, from Quality Software

Typing 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:

```
10 HOME : REM CLEAR SCREEN
```

The LIST will look like:

```
10 HOME : REM CLEAR SCREEN
```

Applesoft inserts spaces into a program listing before and after every command word or mathematical operator. These spaces don't pose a problem except when they are inside of quotes or after a DATA command. 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 special characters (◊). All other spaces are there for easier reading.

NOTE: If you want your checksums to match, only type spaces within quotes or after DATA statements if they are shown as (◊) characters. SAVE the program at periodic intervals using the name given in the article. All characters after a REM are not checked by the checksum program so typing them is optional.

Typing 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. When finished, return to BASIC with:

3D0G

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

Typing Source Code

The source code is printed to help explain a program's operation. To enter it, you need an

"Assembler". Most of the source code in older issues is in S-C Assembler format. If you use a different assembler, you will have to translate portions of the source code into something your assembler will understand.

Computing checksums

Checksums are 4-digit hexadecimal numbers which tell if you typed a program correctly and help you locate any errors. 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 are on the "Starter Kit".

If your checksums do not match the published checksums then the line where the first checksum differs is incorrect.

CHECKSOFT instructions: Install Checksoft (BRUN CHECKSOFT) then LOAD your program. Press & to get the checksums. Correct the program line where the checksums first differ.

CHECKBIN instructions: Enter the monitor (CALL -151), install Checkbin at some out of the way place (BRUN CHECKBIN, A\$6000), and then LOAD your program. Get the checksums by typing the Starting address, a period and the Ending address of the file followed by a ctrl Y. SSSS.EEEE ctrl Y

Correct the lines where the checksums differ.

Writing to the RDEX editor

RDEX (are-decks) stands for: Reader's Data EXchange. We print what you write. 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. If you discover any errors, please let us know so that we may inform our other readers.

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

Because of the great number of letters we receive and the ephemeral and unpredictable appearance of our volunteer staff, any response to your queries will appear only in 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.

How to get a free library disk

Whenever possible, send everything on Apple format (5.25" - DOS/ProDOS or 3.5" - ProDOS) or IBM format (3.5") disks. Other formats are acceptable but there may be some delay as we look for someone to translate it for us. (If you use a 5.25" disk, when we print your letter, we will return your disk with the current library disk copied onto it.) Use whatever text editor you like, but tell us which one. Put a label on the disk with your name (or pseudonym) and address (if you want to receive mail). Don't reformat any programs or include them in the text of your letter. Send Applesoft programs as normal Applesoft files and machine language programs as normal binary files. We have programs to convert them to the proper format for printing. If you are

sending source code files, and you are not using the S-C Assembler, send them as normal text files.

When to include a printed letter

Don't include hardcopy (printout) unless:

- a. You are writing about a bug or other printing error.
- b. You are writing to ask for help.
- c. You are answering another readers help request.
- d. You are writing about your subscription or sending an order for back issues or software.

Bugs, requests for help and answers to requests for help are bumped to the head of the line and go in the very next issue. All other letters are printed in the order that we receive them.

Writing to get help

When writing to request help, be sure to include ALL relevant information. The more information you include, the easier it is to find a solution. There's an old saying that goes "A properly framed question includes 90% of the answer".

How to get mail

If you are interested in receiving mail from other readers, be sure that we have a current address. If you use a pen name and want to receive mail, we need to have your address. Our readers privacy is important, so we will not print your address unless you specifically say too.

How to write to RDEX authors

When writing to one of the RDEX authors. Write your letter and seal it in an envelope. Put your return address, the authors name (as it appears in RDEX) and the correct postage on the envelope. Put this envelope into another and send it to RDEX. We will put the correct address on your letter and mail it for you. Check to the right of the authors name to see if the author is writing from a foreign country and include the proper postage.

Help Line

These readers have volunteered their time to help you. Please call only within the given time frames (corrected for your time zone). No collect calls.

Jack Nissel (Disk Protection, 7-10PM EST)
(215) 365-8160

The BBS

(Bulletin Board System)

Dave Goforth is the sysop for the Computist BBS. The number is: (206) 581-9292. If you already have a User ID# and password, sign-on using the User ID#. If you are a new user, it may take a day or so to validate your new ID# and password.

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

Editorial Notes

Bad GS Power Supplies

I read a note in "Enhance" (Quality Computers in house newsletter and catalog) that says that Apple Computer Inc has determined that certain GS power supplies may fail under low load conditions. That means that the power supply may fail if it isn't loaded enough. How's that for illogical?

If your IIs was manufactured after October 1990 then you are OK. (That's manufacture date not date of purchase.)

To determine if your computer is affected, open the case and take a look at the power supply (the box on the left). If the case is a dull brushed metal, the serial number begins with the letter "I" and there's no red dot on the case you may be in for trouble.

Apple has a repair extension program for the affected power supplies. They will replace failed power supplies thru January 31, 1993 but only failed supplies. There's no way to get a functioning but suspect supply "fixed".

If you already paid for an out-of-warranty replacement supply, you may be eligible for a refund. Contact your nearest Apple dealer before June 30, 1991 or call Apple Customer Assistance at 1-800-776-2333.

IRS news

We are still nibbling away at this one. Thanks to all of you who sent a donation last month. I have a suggestion. If you are considering sending a donation, send it as a back issue or library disk order. That way you get something immediate in return. Kind of like a thank you note.

Centerfolds?

While I was waiting for some more material to fill issue #80, I had a brainstorm. I was in the process of making another "chart" for the wall next to my computer. You know, pasting together 8.5" x 11" sheets to make a larger chart. Anyway, I was thinking that if you pull out the center page of Computist, you would have a 17" x 22" sheet. That would make a nice sized chart. So how about it? Send me copies of all of the "charts" that you make for yourself and I'll turn

them into "centerfolds" in Computist. Send the info on disk if at all possible. If you don't have a chart but you do have an idea for one then send that too.

Postage Increases

You knew it was going to happen and here it is. I finally took some time to calculate the postage increases from the US Postal Service and add them to the subscription rates. Good news and bad news. The regular US and 1st Class rates are going up by one dollar (\$1). Foreign rates stay the same since we use a mail forwarder and they haven't raised their rates yet. So if you are coming up on your renewal, do it now! As of the mailing of issue #81 (the next issue) we will begin charging the new rates. This is your last chance to renew at the old rates.

Help Questions need help!

I receive a lot of help request letters. Some of them will never be answered. Not because I don't have the time, although time is always short here, but because they are of the "open-ended" type of question. They fall into two main categories.

The first one I call the "What is Life?" questions. The ones I receive most often are "What do I use my Apple II for?" and "How can I make some money using my Apple II?" Other popular requests are "Can you help me write a program to do...", "I have lots of memory, can you make my copy of (program name) work with my copy of (program name), in memory, at the same time?", "How can I make my IIE copy of (program name) use the super hires screen on my IIGs?".

The other category is "Incomplete Questions". These are open-ended because of the lack of full information about the system being used. They are usually about software but hardware and firmware questions do pop up. Examples are, "My system locks-up occasionally, why is this happening and what can I do to stop it?", "Sometimes I can't catalog a disk and other times I can. What's happening?", "My copy of (program name) won't run, how can I fix it?", "My copy of (program name) runs on a friends Apple but not on mine, why?".

Asking these kinds of questions is like calling a doctor and saying that you feel bad and asking what is wrong. The doctor is not going to hazard

a guess without asking you a lot of questions. The doctor needs more information and so do I. If you are serious about getting an answer, you must provide complete information. Which Apple are you using? What other hardware is plugged into your Apple? What operating system are you using (DOS3.3, ProDOS, GSOS, GEOS, etc.) and what software is resident in your computer when the problem appears. Everything is important and almost anything might be the one bit of info that we need to answer your question.

Anyway, when it comes to answering question, I'm a little burnt-out. Perhaps someone else would like to answer these questions while I take a break.

Will Computist ever come out ON TIME?

I don't know, maybe not, but I have a real good excuse for issue #80. I caught up on submissions as of issue #79 and didn't have enough to fill issue #80. I could either wait a bit longer for YOU to send in more info or I could reduce the number of pages in the issue. Since Computist is printed on a newspaper web, it is run as three (3) signatures of eight (8) pages each for a total of 24 pages per issue. Reducing the page count would mean going to 16 pages. (Two - eight page signatures.) I decided to wait a bit longer. You don't really mind, do you?

In any case, don't delay, send in your softkeys, APTs, tips and informational articles RIGHT NOW. I'm interested in any BASIC programs (long or short) that you have written and would like to share with the rest of us. I'm especially interested in any ampersand (&) utilities designed to make programming in BASIC easier. Come on! I know most of you have something that you have written and still use. Share it with us.

New Eamon Adventures

There are two (2) new Eamon scenarios. They are "Utterly Outrageous #205" and "Curse of the Hellsblade #206".

Oops!

I forgot to merge the Help file with the main text. That means I've got 6 annoyed readers who sent help requests which won't be printed 'till issue #81. Sorry...

Table of Contents

Editorial Note	3
RDEX Contributors	23
Most Wanted Softkeys	23
unClassifieds	23
The Product Monitor	
Reviews	
AT Bus Design	5
Apple IIGs Hardware Reference, second edition	5
Countdown	4
Genius Mouse:GM-F302/303	5
Logo PLUS	4
Thunderstick	4
Fast Frames, Updates, Etc.	
A Copy II Plus PLUS!	7
Countdown: E-mail	6
Hole 19 Talk	5
Holey Megabytes!	6
Keeping the Wiz in Wiz III	6
Logging ProDOS Drives	6
Next	7
Platinum Yanked!	7
Program Writer	6
Salute!	7
XT-AT Handbook	7
'Courses of 1990	5
'Unlimited Golf& Course Design	5
Why Logo?	4
Vendors	7

Questions:

i 8, 16, 17, 21, 22

Answers:

Edison 21
Groucho 21

Ross A. Holmes	21
E.N. Hondrick	21
Jeff Hurlburt	21
Bob Igo	21
Blain Johnson	21
Gary Wills	21

Features, Notes & such:

Add Copy II Plus file handling to your BASIC progr. See A Copy II Plus PLUS!	7
Comments on the Beginner's Book	7
Formatting 720K disks as 1.44M HD. See Holey Megabytes!	6
How to SAVE hexdumps as CDA's	8
Logging ProDOS Drives	6
The Basics of Kracking (part 7)	17
The Basics of Kracking (part 8)	18

Bitkeys:

Black Magic	16
Guild of Thieves	16
Gunslinger	16
King's Quest Series	16
Leisure Suit Larry	16
Man Hunter: New York	16
Police Quest	16
Realms of Darkness	16
Saracen	16
Sierra Boot Disks	16
Silicon Dreams	16
Space Quest Series	16
Ultima V	16
Wizardry Series	16
Xyphus	16

Softkeys:

Ancient Art of War	9
Battle Chess	10
Bridge 6.0	10
Captain Blood GS	10

Dinosaur Days v1.0	20
Empire	8
Fahrenheit 451	19
Fay's Word Rally	20
GATO v1.3	21
Greeting Card Maker	20
Hostage	15
Keef The Thief	8
Magic Spells v2.0	19
Magic Spells v2.1	19
Mickey's Crossword Puzzle Maker	16
Monsters and Make Believe v1.1	20
Pipe Dream	8
Pipe Dreams	15
Rear Guard	17
Rendezvous with Rama	19
Same or Different	20
Teacher's Tool Kit	19
Teacher's Tool Kit (IIC)	20
War of the Lance	10
Where in the USA is Carmen Sandiego?	9
Windwalker GS	15
Windwalker IIE	15

APTs:

Space Rogue	16
Wizardry III. See Keeping the Wiz in Wiz III	

Playing Tips:

Countdown	6
Space Rogue	16

IBM Softkeys:

Serve and Volley	22
Welltris	22

The PRODUCT MONITOR

RATINGS

Superb	★★★★★
Excellent	★★★★
Very Good	★★★
Good	★★
Fair	★
Poor	☹
Bad	☹☹
Defective	☹*

Countdown

★★★★

\$59.95 for VGA 640K PC
(\$9.95 for Hint Book)

Access Software

Your problem is that you do NOT belong here. The old noodle may be full of mush—something about headlights, a car, a dingy nuthouse cell...—but, you're bound to be one of the 'good guys'. "Mason Powers", the label on the chart, SOUNDS like a 'good guy' name, doesn't it? Now, if you could just figure out where 'here' is and how to get out (and WHO dumped you here, and why, and ...). Well, maybe you'd have a shot at finishing whatever it was this Mason Powers person was doing. Like, maybe it was important?

It was; it IS! Somewhere, a terrorist group called "Black December" has started a countdown to disaster. Bad enough; even worse is that 'December's string of recent 'minor' atrocities AND their planned master stroke may be the work of a mole at the heart of CIA's Middle East section. Safely stashed in a 'company'-run Turkish sanatorium, you are the sole remaining loyal CIA operative who believes the mole exists. The other guy, Mc Bain, is supposed to have been blown away by (who else?) Mason Powers!

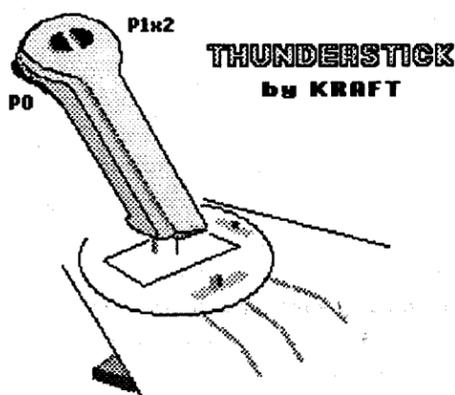
Countdown launches you on a continent-hopping quest to shut down Black December, pull the plug on its operation Thunderbolt, and nail the mole known only as "Stormbringer". Billed as "an interactive movie", the game employs real actors in several VCR-playback-quality animated 'cut scenes' and encounter close-ups—all in 256-color VGA. Digitized voices and full-range sound effects are available via AdLib, Sound Blaster, and other popular adapters.

To maintain its 'in the movie' effect, Countdown's formats adjust to fit your current activity. Except for the surprise ending, cut scenes serve as flashbacks to jog your befogged memory. Most of the time you will have a 3-D side view of your Mason Powers figure in a room. Here, using KB or mouse, you can move around and GET, USE, etc. items. In the sanatorium corridors, Black December headquarters, etc. the view is lower-

detail top-down; a still lower detail view helps with navigating an underground maze.

Many important clues (and some critical items) are obtained in numerous face-to-face encounters with such personages as Lisa, an agency colleague, a Russian agent named Boris, and Golden Desire(!), a dead-ringer for Angie Dickens—to name a few. In these the main challenge is to select the optimum dialogue approach (e.g. "Friendly", "Hassle", etc.) guided by character comments and facial expressions. Other displays include travel maps, city picture inserts, and, for high-tech decoding and data processing chores, your handy Computer Access Device can do blow-ups of photos and messages.

Countdown's best feature is that, having wowed the player with 'special effects', it remembers to deliver a highly-engaging, playable adventure. One CAN win this game without the Hint Book. (I did make one HELP! call to Access, before arrival of the Hint Book—and used the book to get around one problem. In each case the 'answer' was well within reach in the game.) I just KNEW the mole's identity—"It HAS to be ..."—so, there was no way this game was going on the back burner. Expect the promised "intrigue, espionage, and suspense." Once the Countdown starts, it's too late to back out!



Thunderstick

★★★★

(Joystick for Apple II series,
PC version available)

\$25 (approx. retail)

Kraft Systems

Size: 7.0"W x 6.5"D x 2.25"H x 5"
Handle

Centering Tension: approx. 150 gm.

Handedness: none (Right, if using slide
throttle)

Centering Error: 1.0%H, 1.2%V

Centering Defeat: none

Movement Range (Horiz.): 50 degrees

Useful Range (X/ Horiz.): 33 degrees

Centering Adjust: two 'slides'

Connector: Apple DIN (16-pin adapter
available)

Case Access: five screws

Done in steely flat-black, Kraft's new combat stick features a triangular-ish sloped-front case topped by a large, full-grip handle. All buttons are positive click-action type switches. PB-0, the 'fire button', is your trigger. PB-1 is duplicated on the top-of-handle mini-panel for convenient thumb activation by left or right-handers. The "throttle" is a slide control mounted on the case's left edge. Handle position controls outputs of J-0 and J-1 (the normal '1st-player' X and Y analog outputs). Throttle sets J-2 (normally, the '2nd-player' X output).

For two-stick games or games which do not employ a throttle option there is a right-edge-mounted slide switch to disconnect the slide control.

I put Thunderstick through the usual driving and combat tests. These, including "Test Drive II"/gs and "Elite"/II+ played (ahem) with accelerator card, revealed the anticipated flight readiness. (The tab-top X-Y slide adjusters have plenty of range to permit centering at accelerator speeds.) T'sticks hefty slotted-bands mechanism delivers tight, smooth response with enough 'centering-bump' for good stick-position feedback and almost no tendency to 'bottom' (i.e. you can 'feel' center without a 'catch' in handle movement). Needless to say, I've seldom had an easier time potting "Elite"'s Thargoid motherships.

As indicated in the specs listing, "Centering Error" (the tendency to 'pull' X-Y center output in the direction of most recent stick movement) is almost nonexistent. Aside from being a good 'quality measure', low centering error makes for more accurate targeting and enhances survivability in high-action seat-of-the-pants combats. Tension is about right; 150 grams is not at all high for a full-grip stick. As for the throttle, I have not, so far, tried a product which employs this option. A tester program read-out showed smooth response over the full, 0-255, range.

The one spec likely to raise a few eyebrows is "Useful Range". For best control precision, 33 degrees (about +/- 17 degrees from center) IS on the low side of 'ideal'. However, on a full-grip stick, a few degrees translates into a lot more actual motion than is the case with a small, thumb-and-forefinger-grip unit. (In fact, beyond +/- 30 degrees, most full-grip stick users are likely to become uncomfortable with the resulting large movement swings.) Another plus factor is that, with +/- 25 degrees of total movement allowed, very little of the range is inactive. Stick travel end-points are good 'signals' for min/max control outputs.

Thunderstick must be placed on a surface at approximately elbow level (i.e. about where one expects to find a flight console controller). Much higher and you will find your hand repeatedly scrapping the centering adjusters. For best results, allow space for your 'stick arm' to rest and swap-in the optional suction cup feet. Since much of the push on the unit is lateral, the normal rubber feet may not supply enough grip on some surfaces. A second caveat relates to the handle itself. Unlike most large-handle sticks, Kraft's unit is not designed for a wrap-around grip—the trigger winds up under the middle of your trigger finger. Instead, the smooth, squarish handle favors a half-wrap grip to place the trigger where it belongs and your thumb on one of the two PB-1 buttons.

It's easy to see why Baywoof has been so enthusiastic about his PC version of Kraft's new entry in the performance-stick sweepstakes. Great looks and a solid, smooth feel add up to a for-real contender. A slightly deeper, rounder handle would be an improvement, particularly for large-handed users; and, as noted above, placement is very important. When you install a realistic flight controller, Kraft seems to be saying, you've got to put it where it belongs. Does it belong in your cockpit? If the old 'console' checks out as under-equipped, check out Thunderstick.

Why Logo?

Logo (from the Greek word for "thought" or "word") is very good at handling text objects, allows users to define new commands, and encourages production of nicely structured programs. Like BASIC, it is an interpreted, highly interactive language (i.e. you can write procedures and try them without waiting for a compiler-linker). Unlike BASIC and other popular languages, Logo exhibits amazing range in terms of 'abstraction': from "concrete" at lower 'turtle moving' levels through abstruse at its higher 'lists within lists' levels. Full-featured versions (like Terrapin's) offer exceptional error reporting and debugging, floating point math, and capabilities for making machine language calls. It is fair to say you CAN do anything in a good Logo implementation that you can do in BASIC, Pascal, or C; but, almost always, it will take longer.

Logo is very popular as a 1st programming language for young children because they can make concrete things happen using commands that 'say what they do'. In various U.H. CFG projects, I've encountered both BASIC and Logo educational situations. BASIC is generally preferred for middle school grades. Logo appears to work better as a starter language for younger students for whom BASIC's abstract, cryptive commands pose problems. Mainly, adults 'get into' the language for one or more of four reasons: A. to teach computer programming to young children; B. to develop Logo educational materials; C. to research artificial intelligence and "micro world" concepts; D. for fun!

Logo PLUS

★★★★

\$119.95 for 128K Apple II

Terrapin

After years of depending upon Terrapin's DOS 3.3 based "3.0" language for my Logo computing, the arrival of ProDOS based Logo PLUS was a genuine 'event'. Now I could put loads of Logo programs on 800K diskettes, set up handy subdirectories, and REALLY get organized! There are many new built-in commands, too. Called "primitives" in Logoese, these include a set of commands to take advantage of ProDOS (e.g. MAKEDIR, ONLINE, ...), NOTE for playing music, some build-ins of previously user-created commands (e.g. WAIT), and several important Graphics enhancements. Via DOT? SDOT?, and COLORUNDER your turtle can now detect the presence and/or color of dots. FILL colors-in an enclosed area; you can HFLIP and VFLIP the hires screen; and ZOOM is for detailed hires editing.

To the above you can add PRINT-SCREEN for hires screen dumps plus whole new sets of commands for Shape Editing and for placing TEXT on the graphics screen! Among the on-disk utilities you'll find GWRITE, a tool for adding text (in various fonts and styles) to hires screens, EDSHAPE, a separate Shape Editor, a Font Editor, a 6502 Assembler(!), and files to enable color and inverse hires printouts.

With just a little diddling, all of your favorite old-Logo stuff can RUN under the new version. The Logo PLUS CONVERT utility (or Copy II Plus) will move the DOS 3.3 files onto a ProDOS

diskette. They will transfer as .BIN type files, which may cause some concern, if you CAT a directory of Logo Plus programs—they are all TEXT files! No problem. The new Logo saves programs as .TXT type files; but it can load both .BIN and .TXT program files.

Once transferred, it will usually be necessary to remove or modify any user-created procedures having the same name as new primitives (WAIT is an example). Finally, you will need to be sure that file names mentioned in programs match those shortened in the ProDOS move. As before, program names must still end with ".LOGO". Too bad. With only fifteen characters allowed by ProDOS for file names, Terrapin really should scrap this requirement. Going to the .TXT filetype, however, is a good idea. For heavy-duty editing/converting, you can bring any Logo PLUS-saved or "Apple Logo" program into your favorite word processor and save the results in ready-to-go Logo PLUS form.

Terrapin's new Logo is a major upgrade; it is not the long awaited "Iigs Logo". Super-res and Iigs sound are both ignored; and, while Logo PLUS supports larger programs, 128K is a 'drop in the bucket' of what Iigs could handle. The absence of 80-column text is only mildly surprising. Of course, Logo PLUS SHOULD support the denser format, if only because many prospective users find it difficult to take 40-column format seriously. As, chiefly, a list/text-processing language, Logo should be able to display as much text as the host machine allows. On the other hand, an 80-column display is not as critical to ease of program editing in Logo as in other languages. Logo program lines are seldom very long; and, in schools, easy-to-see 40-column displays are preferred for use by the younger students who make up the bulk of classroom users.

Constructing your own procedures, fine-tuning them, and fitting everything together IS entertaining; Logo is one of the few languages people actually use just for the amusement of programming. No doubt much of the fun comes from 'building stuff' and watching it work; but a lot comes in an on-going process of discovery. Logo PLUS starts you fast with the dauntless Turtle, extensive tutorials, references, and examples—there's even an "Instant Logo" option for younger children—but, with over 200 commands, exploring the language's full potential can be as engaging as any role-play adventure. Supplied on both 5.25" and 3.5" media, the best of the Apple II Logos comes with Terrapin's standard enormous loose-leaf manual, a handy Quick Reference Card, and the spiral bound Getting Acquainted with Logo PLUS—not a bad idea!

Genius Mouse: GM-F302/303



\$99.95 for PC XT/AT and PS/2

KYE International

Interface: PC serial or PS/2 mouse port (using PS/2 adapter)

Encoding: Standard opto-mechanical
Resolution: Three modes. Expressed in terms of mouse movement required

to obtain full-screen (horiz.) cursor sweep:

Low- 0.75"/screen

Medium- 1.50"/screen

High- 3.50"/screen

Connectors: DB 9-pin (on mouse cable) plus 25-pin serial adapter; 6-pin PS/2 adapter cable included in 'F303 package

Buttons: Three keys, including standard Left and Right plus Center

Switches: Bottom-of-case slide switch to set Serial or PS/2 mode and toggle in resolution settings

Size: 4.0"L x 2.25"W x 1.2"H

KYE's new mouse is called "genius" because the user can switch resolution 'on-the-fly' right on-the-mouse. For example to switch from "Normal" to "High" speed (i.e. lowest resolution), you hold down the Left key and toggle a slide on the case bottom. When you're in the middle of a graphics design task and need a delicate touch, a Right key-press plus toggle and ZAPPO your mouse is in high resolution! There's no need to call up a special menu display or maintain a "Config", etc. utility in memory.

Rendered in off-white, Genius Mouse is an attractive unit which fits neatly into a matching pocket for convenient storage. While it performed 'as advertised', three design features in particular explain the rating:

1. The three keys are identical in size. This leaves the frequently clicked Left key too small. In extended trials I often found my finger resting on the dividing space between Left and middle keys.
2. The mouse case features a pronounced 'cupcake' style rim along three sides. Side-gripping fingers must rest uncomfortably on the rim or below it and scrapping the pad.
3. 'Click' action is poor. The sample received exhibited approximately 2.5 mm of space between key and stop. To click you must exert force through the entire distance against the unit's 96 gram tension. The result is a slightly mushy feel, missed clicks, and clicking-finger fatigue. I took the unit apart, sliced into the plastic key supports to reduce tension and glued-in shims to pull the keys down closer to the case. This produced a 1mm stroke, 75 gm of tension and a vastly easier, more positive action.

It has been a couple of years since PC users depended upon mouse suppliers for their painter software. After using EA's "Deluxe Paint II: Enhanced", a little experience with KYE's "Dr. Genius" utility makes it clear why. Old Doc Genius's retirement is LONG overdue; the GM package would look better without it. Oddly, you'll have to catalog the diskettes to unearth one of the better software features included. On diskette is a large documentation file packed with valuable details about implementing Genius Mouse (or ANY mouse) in your programs.

Supplied in an attractive plastic case, Genius Mouse comes with manual, connector(s), software (including .SYS and .COM versions of the driver), pocket, and a small plastic-coated mousepad. Everyone can appreciate good looks and smarts; but in mouseville the bottom line has to be user comfort. You use only one mouse; and you use it a lot! Genius Mouse has too many rough edges.

AT Bus Design



\$69.95 - 203 pages

Annabooks

True, as a typical Apple II user/experimenter, I HAVE found myself thinking about possible hardware mods to our PC/AT. (Just add a lead here; run it to an op-amp in slot two; ... etc., etc.) One thing's for sure, when you are serious about constructing an/AT plug-in, you'll want Annabooks' 'Bus Design manual (by E. Solari) close at hand. Incorporating recent Intel and IEEE standards releases, the book focuses upon ISA and extended ISA ("E-ISA") 8/16-bit bus designs.

Though technical references are seldom as 'dry' as is commonly supposed, it was still something of a surprise to open a text crammed with timing diagrams (tables, charts, ...) and find highly-readable discussions of design ideas and considerations. Beginning with a background and overview of the /AT bus, the book covers Memory and I/O address space, Generic bus cycles, Add-on card interaction (including slaves and bus masters), DMA transfers, Refresh, and much more. One chapter describes each of the bus signal lines in detail.

Organization is very good, with good use of shaded 'extra information' boxes to supply handy details. The most notable deficit is the absence of an index. One or two design examples would be welcome, too; but there's no denying that the book performs as promised and beyond. Better yet, as someone who 'knows his stuff' the author doesn't try to 'snow' the reader with arcane terminology. AT Bus Design is a text which, for someone with a little experience in hardware design, can be a real help in developing a PC plug-in design that works.

Apple Iigs Hardware Reference, second edition



\$26.95 - 323 pages

Addison-Wesley Publishing

As PC computists know all too well, there's a good reason why third party publishers are so critical to users of 'generic' /AT's: when you buy one, the documentation you get is strictly 'bare bones' stuff. Meanwhile, Apple and Mac users continue to bask in a steady stream of quality, in-depth publications from Big Green via Addison-Wesley. The

new 'Iigs Hardware Reference (for BOTH earlier and 1MB Iigs) presents beautifully laid-out charts, tables, and easy-reading explanations in large, 8.5" x 11" softcover format. Ten indexed chapters cover Memory space, Shadowing, the Mega II and FPI subsystems, Sound, Peripheral Slots, Connectors, Desktop Bus, Video displays, I/O ports, Battery RAM, Disk interface, Power supply, 65C816 processor, and much more. "Cover" means just that, including timing diagrams, pages of Soft switches, Register descriptions, Instruction sets, AND nine pages of fold-out Schematics!

'Production value' is very good, with ample white space and effective use of font styles and light-dark highlighting. I do, just a bit, miss the color employed in the first edition; but, overall, the update's charts and pictorials are probably clearer and easier to use. Of course, along with most Iigs users, I continue to believe that the hardware and firmware references belong in a single, integrated volume. (See, one really can find something to pick at in ANY product.) Owners of pre-1MB Iigs machines who rushed out and purchased the original 'Hardware Reference are already in good shape. 1MB version owners and early-Iigs hold-outs (if any) need to rush out and grab the 'Second Edition.

Fast Frames, Updates, Etc.

Hole 19 Talk

When I first reviewed Accolade's JNG, the Iigs version got good marks for overall challenge realism but lost points on screen-update speed, putting challenge, and the absence of a course construction utility. A year has not changed this evaluation. Indeed, for Iigs, Mean 18 (also from Accolade) looks better than ever vis-a-vis the competition (including EA's "World Tour Golf").

'Courses of 1990 ★★★

Whether you enjoy Jack Nicklaus' Golf on the Iigs or PC, you will want to add the latest "Add-on Course Disk" (\$21.95) courses to your collection. These include Oak Hill, Royal Troon, and Kemper Lakes. Based upon an hour or so of sampling the holes, you can look forward to some of the best, most interesting challenges yet.

'Unlimited Golf & Course Design ★★

This new Accolade package (\$59.95, for CGA-VGA 640K PC) significantly enhances Jack Nicklaus' golfing by offering both the game AND a fine course designer utility. The game supplies realistic on-the-course views and



Accolade's trademark 'power-bar' golf-swing simulator plus such expected niceties as computer players, practice setups, on-disk record-keeping, and stroke or skins scoring for 1-4 players. Wind and Course Condition variables assure many challenging rounds even if you never construct additional courses of your own.

'Unlimited Golf' is unlimited by speed. The AdLib music is nice, too; but sound effects are minimal. (Actually, I do not remember hearing any. They may be non-existent.) Putting has been embellished via addition of a pre-shot 'wire-frame' overlay showing contours. Holing a putt is, definitely, NOT the give-away shot it is in the IIGS JNG version. Good. On the other hand, your view is still at 'fairway resolution'—not bad; but not adequate for planning a putt. In as much as Accolade seems determined to 'get it right', they might as well finish the job. **Unlimited Golf** needs AdLib sound effects—be sure and include "Mean 18"'s PFTLOOP! water shot sound—and a separate, close-up putting green display.

Holey Megabytes!

Everyone knows that you're supposed to use 3.5" HD diskettes if you want to get 1.4MB of storage; it says so right there in the MEI catalog. The HD jobs have littler particles than the DD types; so you can fit-in lots more bits. Logical. One can almost see thousands of tiny brown HD marbles, each crisply stamped with a 1 or 0 vs. a few hundred gross DD marbles with crowded, smudged markings.

Gorbash and Baywoof were waiting when I walked into the computer room. "How come you don't format your 3.5" diskettes for 1.4MB?", opens Gorbash.

"Because," I reply with patient wisdom, "they're not HD diskettes."

"Why not just punch a hole on the other side and try it?" pipes up Baywoof.

?Punch a hole?! Having just installed our new 3.5" drive in the PC, I had not yet purchased any HD diskettes—never even seen one. So, they have a hole on the top left edge; so what?! Knowledge is knowledge; and, besides, when I'd tried to format a DD diskette for 1.4MB, "X-Tree" had returned a "Bad track 0. Cannot format" error, not "Wrong kind of diskette." It was obvious that the DD 'big marbles' were smudging the data. Mildly annoyed at being confronted by such cloddish ignorance, my response was "That may be the dumbest idea I've heard all day." Well, they persisted and (surprise, surprise) it turns out Baywoof had even brought over a freshly drilled floppy for the "experiment".

Fingering the new hole and suppressing visions of curled shards waiting unseen within the case, I popped in the mutilated diskette. ... In some saner, better organized parallel universe, my grasp of reality was surely sustained and the diskette did not format. In this one, naturally, it formatted without a hitch!

When I related this discovery to Chuck, he wasn't especially amazed and went on to explain that some of his punched DD diskettes formatted better as HD's than a recent batch of MEI HD floppies! So far, I've processed nearly 30 DD diskettes. About half formatted to the full 1.45MB (i.e. with no "bad blocks"). Of the remainder, all but a

couple formatted to at least 1.3MB—still a BIG improvement over 720K.

Unless you own a rather special punch, the best way to 'HD' a 3.5" DD floppy is with a pointy-tipped soldering iron. Drilling may look neater; but there is a chance bits of plastic may get into the case. The iron is fast; and it leaves round, shard-free 'collars' which are easily sliced off using a pocket knife.

Countdown: E-mail

Greetings, Mason. If you're reading this, it means you have activated the hidden "Computist" key combination on your Computer Access Device. Excellent. We, too, are interested in stopping Black December's plot.

What I've done is run a Gamma 5 simulation of your mission based upon probable contacts, locales, and situational trends. The result was just a 0.009 probability of predicting specific encounter dynamics, but 0.773 of accurately listing Items of Special Interest. Since the simulation indicates that you will visit many places packed with objects of no relevance to your mission, the listing may be of real assistance. Good luck!

```
Gamma MK5// BD.MISSION
.SIMISI >>.DUMP Acct: TURDNIL
+0.001 sec
```

cup	bug	various keys
flashlight	crowbar	boots
rag	wine	opr dummy
knife	scissors	passport
cash	dossier	explosives
CAD	tool box	priest's garb
wallet	screwdriver	wire cutters
blueprints	rock	catapult arm
note	stationery	blanket
photo	postcard	lockpick
message	rope	hook
telegram	gear	attache case
tool	cracker	knockout capsules
dart	scalpel	meat hook
bag	pick	

Keeping the Wiz in Wiz III

"THAT piece", Gorbash commented, "has been invaluable." The article to which he referred is John Wiegley's description of how to transfer characters from earlier "Wizardry"'s to "Wizardry III" (*Computist* #51, page 25). Having 'gotten into' Sir Tech's new 'Trilogy' package re-release (64K Apple II version), Gorbash had completed "Knight of Diamonds" only to encounter "Wiz' III"'s infamous 'spirit barrier'. Normally, developed characters do not actually transfer to "Wiz' III". Instead, your "Legacy of Llylgamyn" guys are supposed to be descendants of earlier heroes. Via a Rite of Passage they undergo a kind of 'spirit transfer' to acquire the "memories and skills" (and names) of "Wiz I/II" characters. Fair enough. It HAS been a long, long time since the legendary Knight of Diamonds quest. Alas, spirit transfer technology has a long way to go; The Rite slashes hard-won attributes and rips off possessions.

JW's technique gets your guys into the "Legacy" scenario, too; but, intact! The trick—adding \$20 to the last byte in each character's block on the earlier scenario's diskette—works. The problem is that finding the last byte isn't always such a simple matter. Some blocks are split and the second part may not be in an adjacent sector. To simplify things, we finally went back to "Knight of Diamonds" rested-up/spelled-up the characters, and noted Level, Hit Points

(current & max), plus any mage and priest spell counts (e.g. "9/9/9/9/7/5 ...") and current Armor Class. As shown in Rob Hall's "Wizardry" character block break-down (*Computist* #40, page 25—another invaluable Wiz resource!), these numbers appear in the latter part of the block. Level is at \$84, \$85; Hit Points at \$86, \$87 (current) and \$88, \$89 (max); Mage spells at \$92-\$9F (2-byte counts for each spell level 1-7); Priest spells at \$A0-\$AD; and current Armor Class at \$B0, \$B1.

After exiting "Wiz' II", we booted "Copy II Plus", hopped to the Sector Editor, and used these numbers to track down the second parts of split blocks. For example, suppose Rev Gun's block shows up as starting at byte \$B0 on the Sector Editor display for Track \$1A, Sector \$00. (You see "REV GUN" in flashing letters.) Each "block" is \$D0 bytes long; so, you know \$50 bytes are here and the last \$80 bytes of the block are somewhere else. If Rev Gun ended "Wiz' II" with 283 Hit Points, then (if he left with current HP= max HP), you might SCAN for hex bytes 1B 01 1B 01. Track \$19, Sector \$00 is a safe place to start, since character block storage seems to begin around Track \$1A. When "Copy II" finds the numbers, you can use other numbers to verify that, for sure, this is the rest of Rev Gun's block. Some 'good try' places to look for block continuations include Track/Sectors \$1A/\$0F, \$1B/\$00, \$1B/\$0B, \$1B/\$0C, \$1B/\$0E.

The creators of "Wizardry III: Legacy of Llylgamyn" were concerned with the problem of designing a challenge appropriate for heroes which may come from either "Wiz' I" or "Wiz' II". They have come up with a fairly plausible device for re-leveling the 'playing field'. The question is: just how much re-leveling are you willing to accept? Is the "legacy" of your new characters merely names and memories OR does it also include developed powers and a pile of hardware? It's up to you to decide.

Logging ProDOS Drives

If you develop a ProDOS application which accesses data from two or more drives, it is usually very helpful to have the names of on-line volumes available to your program WITHOUT the necessity that the names be fixed in the program or that someone type them in each time diskettes are swapped. For instance, a super-res picture-packing/display program may have to work with any of several hundred picture storage diskettes, each with a different name (e.g. TITLE.SCREENS.1, TEMP.SAVE, AD-VENTR.PICS.5, etc.). You want the user to be able to display current (on-line) disk names and quickly select Source and Destination volumes. This is much more convenient for the user than having to keep track of each diskette's location and type-in volume names whenever new diskettes are swapped-in.

LOG.DRIVES is a short BASIC routine I wrote when, at last, the bother of typing in volume names exceeded the expected effort involved in changing the program. In fact, thanks to Beagle Bros's **Program Writer** (★★★★, \$49.95 for Apple II series), the effort proved MUCH less than anticipated.

Using the ProDOS PREFIX command, LOG' attempts to read the volume name of Drive 1 and Drive 2 in "slots" 1-7. When a disk (including any RAM disks) is found, its name goes into

the PF\$(0) array. If no disk is located at the slot/drive accessed, an error occurs. The ONERR routine checks to see if the error occurred in the LOG' routine (SS=4); and, if so, sends you back to continue the search without an error message. Otherwise, ONERR displays the error # and the error's line #, does a get-key call, and hops back to your program. (You can set any additional error handling options, set your own 'back to program' address, etc.)

LOG' also keeps track of associated Slot and Drive #'s in SL() and DR(). ProDOS, of course, aims to free users from any need to worry about slots and drives. I ended up deleting the slot/drive-saving instructions from my application. They are included here just in case your application can make use of slot/drive information.

Line 130 is not part of the LOG' routine. It is used to illustrate how the PF\$(0) location can be used to remember the prefix of the program diskette. (PF\$(0) is not updated by the LOG' routine.) Sometimes there is special stuff on the program diskette which the program will want to access long after the diskette has been swapped out. With the prefix saved in PF\$(0), your program can detect a NOT FOUND error and request that the user "Please Insert Disk ..." by name. Similarly, lines 520-540 and 1900 are included so that you can do a RUN and see LOG' at work. One embellishment not shown is a routine to permit user addition of subdirectory names to the volume names. I maintain these user-created pathnames in an enlarged PF\$(0) array at PF\$(8)-PF\$(14). Like PF\$(0), they are not checked or updated by LOG'.

To use LOG.DRIVES just copy the program lines into your application. By far, the easiest approach is to maintain LOG' on disk as a separate BASIC file and use Program Writer's Renumbr and Copy Lines functions to patch it into your applications. For example, a patching session might go something like this:

```
RUN PW.EDITOR.LC
LOAD LOG.DRIVES
&& (to enter PW editor to edit
LOG.DRIVES)
Delete REMS and un-needed stuff;
Renumbr to 9000..., etc. so that LOG'
lines will not overlap lines in your
application
Copy lines to PW 'clipboard'
[Open-Apple Q] (to Quit the Editor)
LOAD GRAFIX.IO or whatever (i.e.
your application)
&& (back to the editor)
Copy LOG' stuff from 'clipboard' into
GRAFIX.IO
Edit as needed (e.g. to hook-up GOS-
UBs, etc.)
[Open-Apple Q] (Back to ProDOS/BAS-
SIC)
SAVE GRAFIX.IO.NEW
```

LOG.DRIVES

```
100 NV = 7: DIM PF$(NV),
SL(NV), DR(NV)
110 ONERR GOTO 3000
120 REM ..next line
remembers startup volume
name
130 PRINT CHR$(4)"PREFIX":
INPUT "": PF$(0)
500 REM ..somewhere in here
you GOSUB 2000 whenever
you wish to log drives
510 REM ..For example:
520 GOSUB 2000
```

```

530 PRINT : PRINT "Volumes
On-Line": PRINT
540 FOR I = 1 TO ZC: PRINT
PF$(I);TAB(20);" (Slot
";SL(I); " Drive
";DR(I);)": NEXT I
1000 REM ..a place where
program normally resumes
after an error (see line
3040)
1900 END
1990 REM ..About LOG DRIVES:
NV value assumes no more
than 7 drives will be on-
line
1991 REM ..Volume names are
in PF$(x).
1992 REM ..For each Volume,
Slot is in SL(x); Drive is
in DR(x)
1993 REM ..Sub returns with
ZC= number of volumes
detected
1994 REM ..SS is var used by
ONERR routine to get you
back to LOG DRIVES
1999 REM ..LOG DRIVES
2000 PRINT CHR$(
(4)"FRE(0)":ZC = 0:SS = .4:
FOR I = 1 TO 7:PF$(I) =
"": NEXT I
2010 FOR I = 1 TO 7
2020 FOR TT = 1 TO 2
2030 PRINT CHR$(
(4)"PREFIX,";I;",";d";TT
2040 PRINT CHR$(4)"PREFIX":
INPUT "":Q$:ZC = ZC +
1:PF$(ZC) = Q$: SL(ZC) =
I:DR(ZC) = TT
2050 NEXT TT
2060 NEXT I
2070 SS = 0: RETURN
2999 REM ..ONERR
3000 TEXT : HOME :ER = PEEK
(222):EL = PEEK (218) +
256 * PEEK (219)
3010 CALL - 3288
3020 IF SS = 4 THEN GOTO
2050
3030 PRINT "Error #";ER;" in
Line ";EL: CALL -756
3040 GOTO 1000

```

XT-AT Handbook ★★★★★

Measuring just 6" x 3.5", the 89-page **Handbook** "for Engineers, Programmers, and Other Serious PC ... Users" (by Choisser & Foster) is available from Annabooks for \$9.95. When it arrived my first thought was "Aha, this is a 'phony price' item designed to be an 'extra' to entice purchasers of more expensive products." (e.g. "Call now and get the Handbook free!") Maybe; but, after numerous productive, time-saving references to this pocket-sized wonder, I'm not so sure. Among the 56 tables and lists you'll find an I/O map, Hardware interrupt addresses with functions, Bios entry points, Bios data area addresses by function, Diagnostic beep and error codes, /AT CMOS bat-RAM addresses, Screen codes, Line drawing character codes, DOS commands summary, Pinouts for Slots, Game control adapter, RS-232C, Display cable, Parallel printer, ... Wow! And all in a booklet you can carry in a shirt pocket to show everyone you're a SERIOUS PC user!! "Small" really is beautiful. The **Handbook** is a fantastic bargain.

Platinum Yanked!

Right. Though I've been using, testing, and liking Beagle Bros's multi-window **Platinum Paint (PP)** since it's arrival, the brutal truth is that the 1.0.3 version is buggy enough to merit pulling the announced review pending receipt of a 1.0.5 update. The 'last straw' was encountering several instances where the new super-res painter simply re-

fused to place text on some portion of an opened, active work space. Other detected glitches include...

Occasions when **PP's** printer interface refused output to a connected Imagerwriter II.

Failure to import valid hires and double-hires pictures: **PP** insists that hires pics be \$2000 long and that double-hires pics be \$4000 long.

Of course, virtually every popular hires and double-hires utility (and everyone else) automatically whacks off the last eight bytes of such blocks when doing a SAVE. (i.e. Hires pics are normally \$1FF8 long, NOT \$2000.) The last eight bytes are not part of the display and cutting them conserves a full sector of DOS 3.3 storage per picture.

Failure to allow access to entire work area in 640 mode: To reach the last few right-most pixels, the user must go to a full-screen (no windows, tool bar, etc.) display.

Beagle is aware of the problems and, with typical Beagle class, has promised to supply purchasers of any buggy versions with FREE update fixes as they become available. There will be a small charge if an update also adds a significant new feature.

A Copy II Plus PLUS!

No, Central Point has not, as far as I know, fixed the RAM disk problems in its **Copy II Plus 9.1** bit copy routines. (I'll let you know when I get an update.) This does not, however, prevent the ongoing discovery of neat applications for the utility. One you may not have considered is incorporating **Copy II's** powerful file-handling capabilities as part of your ProDOS programs!

On the principle that one specific example is worth a hundred generalizations, suppose you've written most of a BASIC program named FLOOTE to edit and play music. (FLOOTE is one of several BASIC programs in the main directory of an 800K ProDOS 1.4 diskette named /NEAT.PROGS.) You want to be able to Rename, Copy, Delete, etc. your music files while running FLOOTE. Writing the routines yourself is a possibility; but you are concerned about conserving program and variable space; and, besides, you want to do music stuff, NOW! This is where **Copy II's** Applications launching function comes to the rescue:

Copy UTIL.SYSTEM, UTIL.MORE, UTIL.APPS, and UTIL.CONFIG from your **Copy II Plus 9.1** diskette to /NEAT.PROGS

Rename UTIL.SYSTEM to something else, like UTIL.C2 to avoid any chance of interfering with the usual BASIC.SYSTEM startup upon booting

Do a -UTIL.C2 to start your **Copy II's** utilities copy and use it to create a subdirectory named GO.FLOOTE. Copy BASIC.SYSTEM to the new subdirectory.

Now, still in **Copy II's**, select Edit.Applications and respond to the prompts to enter a NEW application in the list...

Name: FLOOTE

Command Letter: F

Application Prefix: /NEAT.PROGS/GO.FLOOTE

Application pathname: BASIC.SYSTEM

... select SAVE; and exit Edit.Applications

Quit **Copy II** to Applesoft and write a one-line program:

```

10 PRINT CHR$(4) "PREFIX/":
PRINT CHR$(4) "-FLOOTE"

```

Save the program as STARTUP in the GO.FLOOTE subdirectory.

Load FLOOTE and add a new item to your Main Menu: "Go Copy II" (or "File Handling", etc.). Fix your program so that when "Go Copy II" is selected you execute PRINT CHR\$(4)"-UTIL.C2"

Viola! Now when you're running FLOOTE and want to move a bunch of files into a new PIANO.MUS subdirectory, selecting "Go Copy II" zips you into **Copy II Plus's** powerful utilities. When you've finished, selecting FLOOTE from the Applications directory zips you back into FLOOTE. Delays are short enough so that **Copy II's** functions seem to be, very nearly, part of FLOOTE!

If the above seems excessively easy, then the 'really good news' is that it's close to being a worse-case scenario. Since **Copy II** will launch only SYS type applications, you had to get back to FLOOTE via the BASIC.SYSTEM-plus-STARTUP route. Also you did not interfere with /NEAT.PROGS's normal STARTUP (which might be a favorite launcher for the other programs on the diskette). This meant that a BASIC.SYSTEM copy and a one-line STARTUP had to be placed in a separate subdirectory to obtain the 'part-of-the-program' effect. In short, hooking-up **Copy II's** file handling smarts is, at worst, moderately easy. Give your BASIC application its own diskette and the utilities practically fall into place by themselves!

Salute!

Even as the world sleeps, a few hearty men and women maintain a network of bulletin boards dedicated to the information and enjoyment of computer users everywhere. These are the SysOps! They provide an invaluable service, almost always, at their own cost. So, the next time you log-in for a freebee session, take note of any hints that a little 'extra support' would be greatly appreciated. PM is proud to declare June as "Be Kind to SysOps Month"!

Next

In time for planning Summertime entertainments expect Data East's **Continuum**, Sir-Tech's 'Fuzzballs, Stunt Driver' from Access, Something from Accolade, SSI, and Disney, **Dark Designs** from SoftDisk, and Lucasfilm's 'Secret of Monkey Island (including hand-drawn Treasure Map!). Count on some SERIOUS stuff, too; plus, as usual, 'whatever'.

Vendors

Access Software

atten: Susan Dunn
545 West 500 South, Suite 130
Bountiful, Utah 84010
(800-800-4880)

Accolade

atten: Melinda Mongelluzzo
550 S. Winchester Blvd., Suite 200
San Jose, CA 95128
(408-985-1700)

Addison-Wesley Publishing

atten: Abigail Genuth
Route 128
Reading, MA 01867
(617-944-3700)

Ad Lib

atten: Jill Carette
220 Grand-Allee East, Suite 960
Quebec, QC
Canada G1R 2J1
(800-463-2686)

Annabooks

atten: Chris Choisser
12145 Alta Carmel Ct., Suite 250-262
San Diego, CA 92128
(800-462-1042)

Apple Computer

atten: mailstop 361
20525 Mariani Avenue
Cupertino, CA 95014
(405-996-1010)

Beagle Bros.

atten: Bevey Minarovich
6215 Ferris Square, Suite 100
San Diego, CA 92121
(800-345-1750)

Center for Gifted and Talented

atten: Robert Houston
University of Houston/ University Park
Farrish Hall #123
Houston, TX 77004

Central Point Software

atten: Apple II Products mktg.
15220 N.W. Greenbrier Parkway #200
Beaverton, OR 97006-9937
(503-690-8090)

Club Apple

atten: Glynn Tolar
P.O. Box 5338
Pasadena, TX 77508-5338
(713-478-9998)

Electronic Arts

atten: Lisa Higgins
1820 Gateway Drive
San Mateo, CA 94404
(415-571-7171/orders: 800-245-4525))

Kraft Systems

atten: Madalyn Rutledge
450 W. California Avenue
Vista, CA 92083
(619-724-7146)

Kye International

atten: Jennifer
12675 Colony
Chino, CA 91710
(800-456-7593)

Product Monitor

atten: Jeff Hurlburt
7814 Santa Elena
Houston, TX 77061
(713-645-8680)

Sir-Tech

atten: Sheri Mitchell
P.O. Box 245
Charlestown Mall
Ogdensburg, NY 13669
(800-447-1230/315-393-6633)

Terrapin

atten: Dave McClees
400 Riverside St.
Portland, ME 04103
(207-878-8200)

David R. Hopkins

CO

Comments on the Beginner's Book

To the RDEX editor and all readers: I have a suggestion for the "Beginner's book." There are many readers knowledgeable about softkeying on the Apple (you know who you are), but no one able to take on the full responsibility of writing an entire book. How about each person contributing a single chapter on the topic they know best? They can write about anything they want, as long as it is detailed enough for all of us

'novices,' and it fits a topic you want the book to cover. It does not have to be grammatically perfect or even formatted in an exact manner, just complete enough to be fully understood (with no steps left out and nothing assumed as already known).

All we need to do is determine the topics and chapter order and what each person can contribute, then have someone organize and edit the contributions. If we break the project into digestible modules (like structured programming), we can get this accomplished. When we have a 'beta' edition ready, we could send copies (text files on disk) to our 'professional' contributors for review and suggestions, then to you for final editing and printing. It will take some time, but IT IS WORTH THE EFFORT!

Since I am a 'relative' beginner myself, I might not be able to contribute very much substance to the book, but I would gladly volunteer my time and efforts towards organizing and editing the material.

THIS BOOK IS NEEDED, AND WE ARE THE ONLY ONES THAT CAN DO IT! LET'S DO IT NOW!

How about it readers? Can we finally get started on the Beginners Book? Do you have some time to sit down and write a list of suggested topics that you would like to see covered in the book? Maybe you could even write some words on one of the topics. We can do it if you help!RDEXed

Bob Feigenblatt CA

Ⓢ How do I translate/convert 'Springboard Publisher Clip Art' so that it can be used with a GS word processor or paint program? I have tried Roger Wagner's "Graphic Exchange" to no avail.

Norman Dodge WA

How to SAVE hexdumps as Desk Accessories (CDA's)

If you don't have a ProDOS 16 assembler (MERLIN.16, ORCAM, or APW), you can still enjoy ALOY.CDA (Ancient Land of Y's classic desk accessory).

From the Monitor or other favorite binary editor, type in the hex codes listed for ALOY.BIN. Save the file to disk with the following commands after making sure you have an error free copy. From the Monitor or from BASIC enter the following commands.

CREATE */SYSTEM/DESK.ACCS/ALLOY.CDA, T\$B9

BSAVE */SYSTEM/DESK.ACCS/ALLOY.CDA, A\$2000, L\$xxx, T\$B9

Replace the * with the name of the volume you wish to store your CDA in. Replace the xxx with the length of the file ALOY.CDA in hexadecimal (\$48A).

I hope this explanation will help readers who do not have a ProDOS assembler. Here are the hex listing:

ALLOY.BIN

CALL-151
 2000:03 00 00 00 00 00 00 00 \$D1F1
 2008:79 03 00 00 20 0A 04 01 \$DF96
 2010:00 00 01 00 00 00 00 00 \$CDE3
 2018:00 00 00 00 00 00 00 00 \$DD93
 2020:00 00 01 00 00 00 00 00 \$EFF6
 2028:2C 00 40 00 41 4C 4F 59 \$710E

2030:2E 43 44 41 20 20 41 4C \$BE26
 2038:4F 59 2E 43 44 41 20 20 \$B728
 2040:F2 79 03 00 00 0C 41 4C \$5238
 2048:4C 4F 59 20 45 64 69 74 \$8B2C
 2050:6F 72 15 00 00 00 01 01 \$C7F4
 2058:00 00 8B 4B AB D4 E0 D4 \$9207
 2060:E2 64 E2 E2 30 20 40 03 \$AB13
 2068:8C 8D 00 A0 50 A9 DF 20 \$507E
 2070:66 03 88 D0 F8 20 40 03 \$F95C
 2078:A0 A0 A0 A0 A0 A0 A0 A0 \$A99C
 2080:A0 A0 A0 D4 E8 E5 A0 C1 \$4D96
 2088:EE E3 E9 E5 EE F4 A0 CC \$BB76
 2090:E1 EE E4 A0 EF E6 A0 D9 \$4BFA
 2098:A7 F3 8D 8D 00 20 40 03 \$377E
 20A0:A0 A0 A0 A0 A0 A0 A0 A0 \$071E
 20A8:A0 A0 A0 C3 E8 E1 F2 E1 \$5D63
 20B0:E3 F4 E5 F2 A0 D2 E5 F6 \$D8F7
 20B8:E9 F6 E1 EC A0 F6 B1 AE \$CED7
 20C0:B0 8D 8D 00 20 40 03 A0 \$D4C9
 20C8:A0 A0 A0 A0 A0 A0 A0 A0 \$9499
 20D0:E2 F9 A0 D6 E9 EE E3 E5 \$C41F
 20D8:EE F4 A0 C1 EE E4 F2 E5 \$753F
 20E0:F7 F3 A0 B9 A0 CD C1 D2 \$D745
 20E8:A0 B9 B0 8D 8D 8D 00 A0 \$C4D4
 20F0:50 A9 DF 20 66 03 88 D0 \$66A6
 20F8:F8 20 40 03 8D C1 EE E3 \$5643
 2100:E9 E5 EE F4 A0 CC E1 EE \$DCC6
 2108:E4 A0 EF E6 A0 D9 A7 F3 \$3662
 2110:A0 CD F5 F3 F4 A0 E2 E5 \$8A92
 2118:A0 F0 F2 E5 F3 E5 EE F4 \$2D6B
 2120:A0 E9 EE 00 20 40 03 A0 \$0CF0
 2128:ED E5 ED EF F2 F9 AC A0 \$6201
 2130:E9 E6 A0 E9 F4 A0 E9 F3 \$BC9F
 2138:A0 EE EF F4 AC 00 20 40 \$03F1
 2140:03 8D C9 A0 E1 ED A0 CE \$86D7
 2148:CF D4 A0 F2 E5 F3 F0 EF \$1BDC
 2150:EE F3 E9 E2 EC E5 A0 E6 \$0494
 2158:EF F2 A0 F4 E8 E5 A0 F2 \$F4FF
 2160:E5 F3 F5 EC F4 F3 A0 F4 \$F8EF
 2168:EF A0 F9 EF F5 F2 00 20 \$E8B8
 2170:40 03 A0 F3 F9 F3 F4 E5 \$F012
 2178:ED A1 8D 8D 00 20 40 03 \$49EF
 2180:A0 D4 E8 E9 F3 A0 C3 C4 \$A6ED
 2188:C1 A0 F7 E9 EC EC A0 F2 \$EED4
 2190:E5 F3 F4 EF F2 E5 A0 C8 \$0C9D
 2198:D0 A0 F4 EF A0 E9 F4 F3 \$3191
 21A0:A0 ED E1 F8 E9 ED F5 ED \$EEC6
 21A8:8D 00 20 40 03 A0 E1 EE \$397C
 21B0:E4 A0 E3 E8 E1 EE E7 E5 \$08E4
 21B8:A0 F9 EF F5 F2 A0 C7 CF \$6263
 21C0:CC C4 A0 F4 EF A0 B3 AC \$E08B
 21C8:B0 B0 B0 A0 F0 E3 F3 A0 \$6A57
 21D0:EF F2 A0 ED EF F2 E5 AE \$696B
 21D8:8D 00 A0 50 A9 DF 20 66 \$1805
 21E0:03 88 D0 F8 20 40 03 8D \$8339
 21E8:8D 00 20 40 03 A0 C4 EF \$1D16
 21F0:A0 F9 EF F5 A0 F7 E1 EE \$594A
 21F8:F4 A0 F4 EF A0 E3 EF EE \$4872
 2200:F4 E9 EE F5 E5 BF A0 A8 \$72B0
 2208:F9 A0 EF F2 A0 CE A9 8D \$7A33
 2210:8D 00 20 57 03 C9 D9 F0 \$D9D7
 2218:0E C9 F9 F0 0A C2 30 AB \$9BBC
 2220:68 85 E2 68 85 E0 6B 20 \$2684
 2228:40 03 8C 8D 8D A0 D2 E5 \$5648
 2230:F3 F4 EF F2 E9 EE E7 A0 \$E1EC
 2238:C8 E9 F4 A0 D0 EF E9 EE \$8369
 2240:F4 F3 A0 F4 EF A0 EE EF \$8B14
 2248:F2 ED E1 EC AE AE AE 8D \$3880
 2250:8D 00 AF E4 71 01 8F E6 \$152B
 2258:71 01 AF E5 71 01 8F E7 \$735A
 2260:71 01 20 40 03 A0 C4 EF \$22A7
 2268:A0 F9 EF F5 A0 F7 E1 EE \$D62B
 2270:F4 A0 F4 EF A0 E3 EF EE \$77C3
 2278:F4 E9 EE F5 E5 BF A0 A8 \$FDD1
 2280:F9 A0 EF F2 A0 CE A9 8D \$4582
 2288:8D 00 E2 30 AF EF 71 01 \$77AC
 2290:C9 0B 90 10 F0 02 B0 3F \$7DD8
 2298:AF EE 71 01 C9 B8 90 04 \$B538
 22A0:F0 35 B0 33 20 40 03 C3 \$4737
 22A8:E8 E1 EE E7 E9 EE E7 A0 \$0F65
 22B0:C7 EF EC E4 A0 F4 EF A0 \$86DA
 22B8:B3 AC B0 B0 B0 A0 F0 E3 \$44C4
 22C0:F3 AE AE AE 8D 8D 8D 00 \$5B63

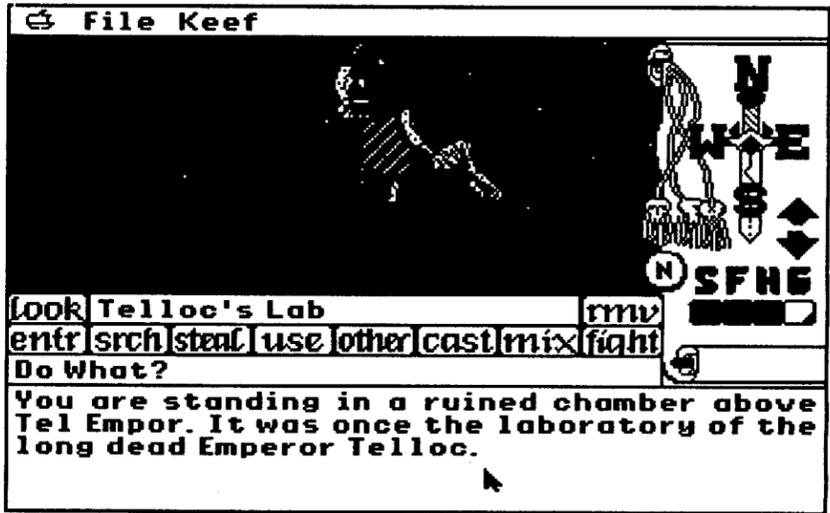
22C8:A9 B8 8F EE 71 01 A9 0B \$BC15
 22D0:8F EF 71 01 4C C1 02 20 \$2B9C
 22D8:40 03 8D D9 EF F5 A0 E1 \$941F
 22E0:EC F2 E5 E1 E4 F9 A0 E8 \$C045
 22E8:E1 F6 E5 A0 B3 AC B0 B0 \$FD85
 22F0:B0 A0 C7 EF EC E4 A0 F0 \$21F7
 22F8:E3 F3 A0 EF F2 A0 ED EF \$04AC
 2300:F2 E5 AE 8D 8D 00 20 40 \$2451
 2308:03 A0 C3 E8 E1 F2 E1 E3 \$6F1B
 2310:F4 E5 F2 A0 E9 F3 A0 E8 \$0437
 2318:E5 E1 EC E5 E4 A0 F4 EF \$8739
 2320:A0 EE EF F2 ED E1 EC A0 \$1D03
 2328:C8 E9 F4 A0 D0 EF E9 EE \$3FB6
 2330:F4 F3 A0 E1 EE E4 A0 00 \$7AE7
 2338:20 40 03 EE EF F7 A0 E8 \$4167
 2340:E1 F3 A0 B3 AC B0 B0 B0 \$41E8
 2348:A0 C7 CF CC C4 A0 F0 E3 \$587D
 2350:F3 A0 EF F2 A0 ED EF F2 \$B8C0
 2358:E5 AE A0 8D 8D 00 20 40 \$C5D5
 2360:03 8D A0 D0 F2 E5 F3 F3 \$A102
 2368:A0 E1 EE F9 A0 EB E5 F9 \$A680
 2370:A0 F4 EF A0 E3 EF EE F4 \$1C61
 2378:E9 EE F5 E5 AE 8D 00 20 \$CF60
 2380:57 03 4C D8 01 08 C2 21 \$8711
 2388:A3 02 1A 83 02 E2 30 A0 \$D97B
 2390:00 B3 02 F0 05 20 66 03 \$A738
 2398:80 EC 28 60 C2 30 AF 00 \$5F29
 23A0:C0 E0 10 F8 8F 10 C0 E0 \$1CF0

code from the code wheel. To find the secret code, you look at the large symbol on the screen and find the matching symbol on the outside edge of the wheel. If you keep pressing the return key or joystick button, the drive will turn on then load the game. After about 3 attempts to play the game or if you get a high score. A message comes up saying you entered the wrong code three times in a row when you started and the program locks up. This patch bypasses the input from the secret code wheel. The other patch removes the music from the Pipe Dream logo which will remove it from the start of the game.

Trk	Scr	Byte	From	To
0C	0D	00	4AF5	EAEA
		09	20EFB4	EAEAEA
		10	4CAA B3	EAEAEA
0C	0E	8C	4C4A55	EAEAEA
		FF	4C	EA

To remove the music from Pipe Dream logo.

Trk	Scr	Byte	From	To
0C	0C	83	2036B6	EAEAEA
			4C0CB6	EAEAEA
			2030B6	EAEAEA



23A8:E2 30 60 48 5A DA 08 C2 \$F2A4
 23B0:30 48 A2 0C 18 22 00 00 \$7349
 23B8:E1 28 FA 7A 68 60 F5 02 \$72FB
 23C0:00 51 03 66 03 F5 02 00 \$BB66
 23C8:3E 03 D8 01 F5 02 00 3B \$8568
 23D0:03 57 03 F5 02 00 1A 03 \$68DE
 23D8:40 03 F5 02 00 F4 02 40 \$D6C9
 23E0:03 F5 02 00 C2 02 40 03 \$39DE
 23E8:F5 02 00 93 02 40 03 F5 \$F31A
 23F0:02 00 90 02 C1 02 F5 02 \$CA4B
 23F8:00 60 02 40 03 F5 02 00 \$B616
 2400:1E 02 40 03 F5 02 00 E3 \$2FDB
 2408:01 40 03 F5 02 00 CE 01 \$2291
 2410:57 03 F5 02 00 A6 01 40 \$88E9
 2418:03 F5 02 00 A0 01 40 03 \$EF71
 2420:F5 02 00 9A 01 66 03 F5 \$5D8B
 2428:02 00 66 01 40 03 F5 02 \$F506
 2430:00 39 01 40 03 F5 02 00 \$1BB2
 2438:2B 01 40 03 F5 02 00 FA \$6091
 2440:00 40 03 F5 02 00 E0 00 \$9704
 2448:40 03 F5 02 00 B5 00 40 \$1913
 2450:03 F5 02 00 AF 00 66 03 \$1732
 2458:F5 02 00 80 00 40 03 F5 \$8F8B
 2460:02 00 59 00 40 03 F5 02 \$9FC5
 2468:00 31 00 40 03 F5 02 00 \$F3D0
 2470:2B 00 66 03 F5 02 00 21 \$615A
 2478:00 40 03 F5 03 00 11 00 \$D4CB
 2480:E1 01 F5 03 00 0D 00 15 \$DA62
 2488:00 00 \$F656

Terry Waskowich

Softkey for...
Keef The Thief
 ?

Requirements:
 Blank Disk 3.5"
 Fastcopier 3.5"
 Blockwarden/Block Editor

With Keef of Thief, I discovered that it had a word protection right after it showed the game picture. The following will show you a quick way to be able to choose any answer and the program with thick it was correct and play the game.

Step-by-step
 1. Make a copy of the game disk.
 2. Edit the copy (Block Warden [F]ollow /KEEF1/KEEF.SYS16 on REL BLOCK 200).

Blk	Byte	From	To
\$3BB	\$167	F00382	800382
	\$175	D00D22	800D22

3. Write the edits back to the copy.
 This patch allows any answer for the word protection to be correct. The game will play just like the original.

Softkey for...
Empire
Interstel

Requirements:
 Blank Disks (5.25")
 Sector Editor
 Fastdisk Copier (5.25")

Big Al MD

Softkey for...
Pipe Dream
Lucasfilm
 Pipe Dream by Lucasfilm has a manual protection scheme. The player must use the large symbol to get the secret

After booting Empire I found that it had a word protection. Instead of the normal word protection, this one also allows you to type in DEMO to see a demo of the game. After finding the protection on the boot disk, I decided to see the demo. After exiting the demo game, Empire reloaded the word protection off the program disk and you had to find it once again.

Step-by-step

1. Make a copy of the three Disks: Boot, Program, and Maps.
2. Make the following edits:

On the boot disk:

Trk	Sct	Byte	From	To
\$1D	\$09	\$80	20 0E 1A F0	20 0E 1A 80
			2B	2B

On the program disk:

Trk	Sct	Byte	From	To
\$06	\$09	\$80	20 0E 1A F0	20 0E 1A 80
			2B	2B

You're Finished! Now you can enter in anything for the word protection to play the game and still be able to enter in DEMO to run the demo.

Softkey for...

Where in the USA is Carmen Sandiego?

Broderbund.

Requirements:

- 512K Apple IIGS
- 3.5" disk copier
- 3.5" disk editor
- 3.5" blank disk

First off, Carmen USA is really neat to play, just wait 'til you get to the higher cases. The Carmen Sandiego series are known as an educational programs, but they are fun for all. Broderbund's programmers took a lot of time in making the pictures and background sounds and if you have played a case, you would already know that. The program is a two disk set, however disk one will hold all the edits you need to do to take out the CP.

The following readings will help you to understand how this protection works.

Make a copies of both disks, you can use any 3.5" disk copier. Make sure you have a 3.5" disk editor handy. Before booting the program you should get into ProDOS BASIC and type:

CALL-151 Enter the monitor
Brings up two hidden CDAs: Visit Monitor & Memory Peeker

You can hit Control-OpenApple-ESC to make sure they are there, now quit back to the monitor. Put the Carmen USA boot disk in the drive and launch it (C500G, PR#5, system restart, which ever you prefer).

Enter your name when asked so you can start a new case. Alright, GUMS-HOE, finish your first case. Once you have finished your case, the wording for the CP will be discovered. The CP doesn't come up until you have solved a case or earned a promotion to be exact. Answering the question correctly will enable you to receive the promotion. The following quotes are used by the WP.

Quote #1: "Before you are promoted you have one more clue to unravel."

Quote #2: "Use the Fodor's USA travel guide to help you answer the following question:"

Quote #3: "That is incorrect. Please try again."

Quote #4: "Sorry, your promotion is denied because you did not unravel the final clue."

Quote #5: "Good job, ..."

Well, now you know the words that the CP uses, get into your desk accessories menu (control panel) by hitting Control-OpenApple-ESC and select Visit Monitor and press return. It'll say "Welcome... Control-Y Return exits" and you are now in the machine monitor.

To find which bank the program is in, look from bank 03/ to 06/ at address \$0000 and look for a \$48 (PHA), once you find it, you know which bank the program is loaded in. 04/0000 return will show you the byte at that location.

In this case, it was 04/0000:48 and so bank 04/ has the main program. To be able to find the wording of quote #1 you have to use some of the features the IIGS has built in. The word "Before" in Quote #1 converts to 42 65 66 6F 72 65 in HEX. Using the P (search pattern) command we'll search bank 3 from 0000 to FFFF (every byte in the bank). Type: **W2 65 66 6F 72 65<03/0000.03/FFFF**

You should see a "03/208E:" appear, this shows where the pattern begins. Type: 03/208D and hit return a couple of times, you will notice the words used in the CP. Keep hitting return to see the rest of the text used for the CP. After looking at that area I discovered at:

- 03/208E: is Quote #1
- 03/20CA: is Quote #2
- 03/2164: is Quote #3
- 03/2117: is Quote #4
- 03/218A: is Quote #5

Now we need to look for references to 8E 20 (\$208E), do this by typing: **W8E 20<04/0000.04/FFFF**

You will see a "29E6:" pop up, this address is where the bytes were found. So type 04/29E2L to list the code found there. (see figure 1.)

04/29E2L

From this you get a feel of of how the program gets its words from bank 03. Now just keep on Listing locations further on, I did, and when I got to this location 04/2AB8 it looked interesting. (See figure 2.)

04/2AB8L

Lets take a look at 04/2B06. (See figure 3.)

04/2B06L

The code in figure 4 is where WP entry and WP correct word is equal.

04/2B5EL

Now I hope this helped you understand how this protection worked. There are at least two different ways of taking out the (WP) on it. First: at address \$04/2AC8 change from D0 09 to EA EA, this allows any entry to be correct. Second: at Address \$04/29E8 change from 22 F0 4D 04 to 22 CA 2A 04. Then right before it prints Quote #1, this new JSL to 04/2ACA will bypass the asking of the WP question and then proceed as if the WP question was answered correctly.

However, after talking with Brian A. Troha, we concluded that the game uses a jump table for most of the routines. If you look at the code from 4/2ACA where it does the LDA #05A0, STA 0D3F, this is where the program sets up to go to the "Good Job," section of code. After some study, we found the actual point where the program would store the WP (A9 96 0A - LDA #0A96 value and changed it to the "Good Job," value. The results are as soon as you have complet-

Figure 1. Listing from address \$04/29E2

29E2:F4 03 00	PEA	0003	32 bit address pointer to address
29E5:F4 8E 20	PEA	208E	03/208E, or the location of Quote #1.
29E8:22 F0 4D 04	JSL	044DF0	prints it up to screen.
29EC:7A	PLY		
29ED:7A	PLY		If you push, you have to pull
29EE:22 EA 4B 04	JSL	044BEA	
29F2:F4 03 00	PEA	0003	Another address pointer, this time to
29F5:F4 CA 20	PEA	20CA	03/20CA, the location of Quote #2.
29F8:22 F0 4D 04	JSL	044DF0	print to screen.
29FC:7A	PLY		the following is more code
29FD:7A	PLY		of the (WP).

Figure 2. Listing from address \$04/2AB8

2AB8:48	PLY		
2AB9:F4 03 00	PEA	0003	bank 03
2ABC:F4 70 64	PEA	6470	address 6470, entry stored.
2ABF:22 6B A2 04	JSL	04A26B	I think this gets the correct word.
2AC3:7A	PLY		
2AC4:7A	PLY		
2AC5:7A	PLY		
2AC6:7A	PLY		
2AC7:A8	PHA		
2AC8:D0 09	BNE	2AD3 (+09)	word & WP entry not equal
2ACA:A9 A0 05	LDA	#05A0	(this area is if the password was).
2ACD:8D 3F 0D	STA	0D3F	(entered correctly)
2AD0:82 8B 00	BRL	2B5E (+008B)	Answer is Correct!, now goto 04/2B5E.
2AD3:AD 63 0D	LDA	0D63	loads Attempt number (1 attempt =100)
2AD6:18	CLC		(2 attempts = 200 & 3 attempts = 300)
2AD7:69 00 01	ADC	#0100	Adds 100 to itself(attempts).
2ADA:8D 63 0D	STA	0D63	stores attempts.
2ADD:AD 63 0D	LDA	0D63	loads the attempts.
2AE0:C9 00 03	CMP	#0300	compares attempts to 300.
2AE3:F0 21	BEQ	2B06 (+21)	not equal goto 04/2B06 = quote #3.
2AE5:90 1F	BCC	2B06 (+1F)	less then 300 goto 04/2B06
2AE7:F4 03 00	PEA	0003	bank 03/ (on 3rd attempt is wrong).
2AEA:F4 17 21	PEA	2117	address 2117. Same as Quote #4.
2AED:22 F0 4D 04	JSL	044DF0	print to screen.
2AF1:7A	PLY		the rest you don't need to know
2AF2:7A	PLY		it denies his promotion.

Figure 3. Listing from address \$04/2B06

2B06:F4 03 00	PEA	0003	bank 03/
2B09:F4 64 21	PEA	2164	address 2164. Same as Quote #3.
2B0C:22 F0 4D 04	JSL	044DF0	prints to screen.
2B10:7A	PLY		the rest you don't need to know
2B11:7A	PLY		it goes back and asking the question again.

Figure 4. Listing from address \$04/2B5E

2B5E:82 13 06	BRL	3174 (+0613)	uses what 2ACA did, and comes back.
2B61:F4 03 00	PEA	0003	bank 03/
2B64:F4 8A 21	PEA	218A	address 218A. Same as Quote #5.
2B67:F4 00 00	PEA	0000	the rest you don't need to know
2B6A:7B	TDC		it goes on as if you have entered
2B7B:18	CLC		the (WP) question correctly.

ed the case(s) for a promotion the program jumps right to the "Good Job," text and continues as though there was never any WP in the first place.

Step-by-step

1. Make a copy of both Carmen USA disks.

2. Make the edit to the copy of disk 1.

Blk	Byte	From	To
\$223	\$17	A9 96 05	A9 A0 05

3. Write the edit back to the copy.

Softkey for...

Ancient Art of War

Broderbund

Requirements:

- Fast Disk Copier 5.25"
- Sector Editor
- 1 Blank Disk

Making a deprotected backup copy of Ancient Art of War (AAW) can be simple if you know a little about the copy protection. When booting the orig-

inal disk, it will show a title page, play music and then runs the program. Make a copy of the AAW boot disk, using a 5.25" fast disk copier or any simple disk copier. Now boot the copied AAW boot disk, you will notice as soon as it shows the hi-res screen it will reboot. This is something to keep in mind. We could look for a section of code that turns on the hi-res screen and the protection shouldn't be to far away.

It's time to start working with the copy, so get out your favorite sector editor. Now since we know it shows a hi-res page, scan the disk for HEX 50 C0. I found it on track 05, sector 00, at byte \$16. We could put a \$00 at byte \$15 to bomb it out when the screen comes up. However, I decided to look at the whole sector and found a \$4C (JUMP) routine at byte \$00. So change byte \$00 from 4C to 00 and write it back to the disk. When the program gets to that location, it will crash and tell us where the routine begins. It's now time to boot

the disk, it should have crashed at address \$4000. (See figure 5.)

4000:4C put back the \$4C
4000L to list the code at \$4000

So lets get back into our sector editor. Now read Track 05, Sector 00 and change byte \$00 back to 4C (from a 00) and write it. Now scan the disk for a JSR 4000 (20 00 40) starting at Track 00, Sector 00. You should find it on Track 00, Sector 01, at \$9D. Looking at the sector starting at byte \$8C we have the following. (See figure 6.)

The call from \$80A0 is to the copy protection routine and can be bypassed by NOP'ing out the call, but keep in mind the carry flag has to be cleared to skip over the reboot call. Now to make AAW think the nibble count has passed and always continue, do the following:

Step-by-step

1. Make a copy of the AAW.
2. Edit the copy.

Trk	Sct	Byte	From	To
\$00	\$01	\$A0	20 08 82	EA EA 18

3. Write the sector back to the copy

You should now have a completely deprotected backup of Ancient Art of War!

Softkey for...

Battle Chess

Interplay

Requirements:

- 5.25" disk copier
- 5.25" sector editor
- 1 Blank Disk

Finally a really neat animated chess game for the Apple IIe: Battle Chess. This is a great game if you like to have your Queen eaten alive, or if you like to see some very creative fighting techniques. You can also play 3-d, 2-d and even over the modem. Other features include: save and load games, change level of experience, and take back moves!

There is one problem with the game, and that is its copy protection, which is word protection. The words are in the back of the manual and are actually opening moves from chess games. Before you can play your first game (get to side b) you must answer a question from the CP, but no more!

Step-by-step

1. Boot up your original Battle Chess, when the menu comes up it will ask if you want to play or make a copy. Make a Copy....

2. After you have made a copy, load up your favorite sector editor, and make the following changes to bypass the word protection and go straight into Side B.

Trk	Sct	Byte	From	To
\$19	\$05	\$40	20 00 B0	AD 00 B0

3. Write the sector back to the copy.

That's it.

Softkey for...

Bridge 6.0

Artworx.

Requirements:

- Apple II w/64K
- 1 Blank Disk
- Sector Editor
- COPYA

Deprotecting Bridge 6.0 is somewhat simple after some inspection. The copy protection was an altered DOS with the epilogues changed to FF FF. Below are

Figure 5. Listing from address \$4000

4000:4C 06 40	JMP	4006	jumps to the routine, to show hi-res.
4003:4C 63 40	JMP	4063	plays music and is executed at \$801E.
4006:8D 10 C0	STA	C010	
4009:20 03 42	JSR	4203	
400C:8D 54 C0	STA	C054	switch to Page 1.
400F:8D 52 C0	STA	C052	show a full screen.
4012:8D 57 C0	STA	C057	to show the hi-res page.
4015:8D 50 C0	STA	C050	switch to graphic mode.
...			
4062:60	RTS		ends the routine & returns to \$80A0

Figure 6. Listing from address \$808C

808C:A9 04	LDA	#04	
808E:A0 40	LDY	\$40	
8090:20 00 8B	JSR	8B00	
8093:20 B5 B1	JSR	B1B5	
8096:A9 05	LDA	#05	
8098:A0 50	LDA	#50	
809A:20 00 8B	JSR	8B00	
809D:20 00 40	JSR	4000	routine to show title and play music.
80A0:20 08 82	JSR	8208	routine to check for a nibble count.
80A3:90 03	BCC	80A8 {+03}	if nibble count is found branch to \$80A8
80A5:4C 91 81	JMP	8191	reboots Slot 6.
80A8:A2 05	LDX	#05	start of the program.

Figure 7. Listing from address \$471C

471C:20 58 FC	JSR	FC58	clears the screen for the question.
471F:20 C8 0D	JSR	0DC8	
4722:20 4B 44	JSR	444B	
4725:20 60 A9	JSR	A960	

Figure 8. Listing from address \$47A1

47A1:20 67 AE	JSR	AE67	
47A4:A5 9D	LDA	9D	
47A6:D0 03	BNE	47AB {+03}	if answer is correct branch to \$47AB.
47A8:4C C2 47	JMP	47C2	re-run word-protection.
47AB:20 58 FC	JSR	FC58	clears the screen when answer is right.
47AE:20 C8 0D	JSR	0DC8	
47B1:20 3D 44	JSR	443D	
47B4:20 10 A8	JSR	A810	
47B7:A9 CF	LDA	#CF	
47B9:A0 43	LDY	#43	
47BB:20 F9 EA	JSR	EA F9	
47BE:20 1A A8	JSR	A81A	
47C1:60	RTS		

the quick steps to deprotect the program using COPYA:

Step-by-step

1. Boot your DOS 3.3 system disk.
2. Tell DOS to ignore epilogos and use COPYA to copy the disk.

LOAD COPYA

CALL-151

B925:18 60

B988:18 60

3D0G

RUN

3. Edit the copy.

Trk	Sct	Byte	From	To
\$00	\$03	\$35	FF	DE
		\$3F	FF	AA
		\$91	FF	DE
		\$9B	FF	AA

4. Write the sector back to the disk.

You should now have a deprotected version of Bridge 6.0.

Softkey for...

Captain Blood GS

Mindscape.

Requirements:

- 512K Apple IIgs
- 3.5" disk copier
- 3.5" disk editor

Here is a quick way of taking the word protection off this game. The word

protection comes up right before you are allowed to play the game.

Step-by-step

1. Make a Copy of Captain Blood GS with any disk copier.
2. Use a disk editor like Block Warden or Copy II+ to edit the copy. (Note: Block Warden [F]ollow /CAPTAIN-BLOOD/BLOOD.SYS16 on REL BLOCK 167.)

Blk	Byte	From	To
\$AE	\$B7	22 F6 05 06	AF F6 05 06
	\$C3	22 2D 06 06	AF 2D 06 06
	\$CF	22 F6 05 06	AF F6 05 06
	\$D9	22 AD 04 06	AF AD 04 06
	\$E4	22 F6 05 06	AF F6 05 06
	\$F0	22 2D 06 06	AF 2D 06 06
	\$FC	22 F6 05 06	AF F6 05 06
	\$106	22 AD 04 06	AF AD 04 06
	\$120	22 E3 00 00	AF E3 00 00
	\$142	22 4B 03 08	AF 4B 03 08
	\$160	22 D7 02 08	AF D7 02 08
	\$189	F0 03 82 5A	EA EA 82 5A

3. Write the block back to the copy.

Softkey for...

War of the Lance

SSI/TSR

Requirements:

- Apple II w/64K
- 5.25" disk copier
- 5.25" disk editor

How to take the word protection completely out of War of the Lance (WOTL) from SSI. To start off we need to make a copy of WOTL boot disk.

Boot the disk and wait until the credit page for the game is shown. Hit CTRL-RESET and WOTL will reset into SSI's DOS, which is called RDOS. What I did, was to see if WOTL was running a basic file to launch the game, by typing LIST at the prompt and I found the following:

```

1 HIMEM: 38655
2 HOME:CALL 3528:A =
  PEEK(9):C=16384:B=1:ONERR
  GOTO 20
3 IF A = 21 THEN & RUN"U",
  16384
4 IF A = 26 THEN & RECALL
  "AP", 17000:& RECALL
  "NAT.DAT",58501 : & RECALL
  "Q1.DAT",25600 :& RECALL
  "Q2.DAT",32600
5 IF A = 16 THEN & RECALL
  "P" + STR$(56138),18000:
  C=19100
9 & RECALL CHR$(A+64),C
10 CALL C:B=2
15 & RUN "V1",38000

```

There were other lines in the program, but nothing that would help us in deprotecting it. At this point I was wondering what value were returned from the PEEK(9) statement and stored in A. So at the Applesoft prompt I typed PRINT PEEK(9), the computer returned a value of 26. Knowing A equals 26, and by looking at the basic program, you can see that lines 3 and 5 will not be used; yet. Continuing, looking at line 10 we see a CALL C which will launch the program. To see what the value of C is, look at line 2 where C is set to 16384. Converting 16384 from decimal to a HEX, we end up with a value of \$4000. Now we need to enter the monitor so we can do some code snooping, to do this type CALL-151

You should start listing code from \$4000 However, since I ran the program with the word protection still intact, I remembered the program cleared the text screen just before the word protection came up. So I searched for 20 58 FC, which is the machine language version of the Applesoft BASIC "HOME" statement. I found the string at memory location \$471C. (See figure 7.)

After doing some checking around, I realized that everything seems to run in order starting from \$4000. We are getting closer, now it's time to look for some kind of conditional branch in this area and I finally found a BNE at \$47A6. (See figure 8.)

From here you can see that \$47AB is the location where the program comes if the answer was right, and eventually returns to the Basic program so it can continue running the game.

Now it's time to make the patch to disk, so let's get out a sector editor. Scan the disk for 20 58 FC 20 C8 0D 20 4B, I found it in two places, change the 20 58 FC to 4C AB 47 to remove the copy protection.

Step-by-step

1. Make a copy of the WOTL game disk.
2. Make the following edits:

Trk	Sct	Byte	From	To
\$1C	\$07	\$1C	20 58 FC	4C AB 47
\$1D	\$0E	\$1C	20 58 FC	4C AB 47

3. Write the edits back to the copy.

Now when booting the newly made copy, it will show the credit page and after hitting the space bar it will auto

Address	Name	Purpose
C000	KBD CLR80COL	Read keyboard. A read to this location returns with the value of the last key pressed. Bit 7 = 1 if value is valid. Disable 80 column store
C001	SET80COL	Enable 80-column store
C002	RDMAINRAM	Read from main 48K RAM
C003	REDCARDRAM	Read from alternate 48K RAM (auxiliary card memory)
C004	WRMAINRAM	Write to main 48K RAM
C005	WRCARDRAM	Write to alternate 48K RAM (auxiliary card memory)
C006	SETSLOTXROM	Use ROM on cards
C007	SETINTXROM	Use internal ROM
C008	SETSTDZP	Use main zero page/stack
C009	SETALTZP	Use alternate zero page/stack
C00A	SETINTC3ROM	Enable internal slot 3 ROM
C00B	SETSLOT3ROM	Enable external slot 3 ROM
C00C	CLR80VID	Disable 80 column hardware
C00D	SET80VID	Enable 80 column hardware
C00E	CLRALTCHAR	Normal LC, flashing UC
C00F	SETALTCHAR	Normal inverse, LC; no flash
C010	KBDSTRB	Turn off keypressed flag
C011	RDLCBNK2	Bit 7 = 1 if LC bank 2 is enabled
C012	RDLCRAM	Bit 7 = 1 if LC RAM read enabled
C013	RDRAMRD	Bit 7 = 1 if reading alternate 48K
C014	RDRAMWRT	Bit 7 = 1 if writing alternate 48K
C015	REXROM	Bit 7 = 1 if using internal ROM
C016	RDALTZP	Bit 7 = 1 if slot zero page enabled
C017	RDC3ROM	Bit 7 = 1 if slot 3 (\$C300) space enabled
C018	RD80COL	Bit 7 = 1 if 80 column store

Address	Name	Purpose
C019	RDVBLBAR	Bit 7 = 1 if not VBL
C01A	RDTEXT	Bit 7 = 1 if text (not graphics)
C01B	RDMIX	Bit 7 = 1 if mixed mode on
C01C	RDPAGE2	Bit 7 = 1 if txtpage2 switched in
C01D	RDHIRES	Bit 7 = 1 if hires in on
C01E	ALTCHARSET	Bit 7 = 1 if alt char set in use
C01F	RD80VID	Bit 7 = 1 if 80 col hardware on
C020	reserved	
C021	MONOCOLOR	monochrome/color selection byte. MONOCOLOR bits defined as follows: bit 7 = 0 enables color, 1 disables color. Bits 6, 5, 4, 3, 2, 1, 0 must be 0.
C022	TBCOLOR	Text/background color selection register. TBCOLOR bits defined as follows: Bits 7, 6, 5, 4 = text color number. Bits 3, 2, 1, 0 = background color number. Colors are: \$0 = black \$1 = deep red \$2 = dark blue \$3 = purple \$4 = dark green \$5 = dark gray \$6 = medium blue \$7 = light blue \$8 = brown \$9 = orange \$A = light gray \$B = pink \$C = green \$D = yellow \$E = aquamarine \$F = white
C023	VGCINT	VGC interrupt register. VGCINT bits defined as follows: Bit 7 = 1 if interrupt generated by VGC. Bit 6 = 1 if 1-second timer interrupt. Bit 5 = 1 if scan-line interrupt. Bit 4 = 1 if external interrupt (forced low in Apple IIgs). Bit 3 must be 0. Bit 2 = 1-second timer interrupt enable. Bit 1 = scan-line interrupt enable. Bit 0 = external interrupt enable (can't cause an interrupt in Apple IIgs).
C024	MOUSEDATA	x or y mouse data register. MOUSEDATA bits defined as follows: Bit 7 = button 1 status if reading x data. Bit 6 = button 0 status if reading y data. Bit 5 = sign of delta 0 = '+' — 1 = '-'. Bits 4, 3, 2, 1, 0 = delta movement.
C025	KEYMODREG	Key modifier register. KEYMODREG bits defined as follows: Bit 7 = openapple key active. Bit 6 = closed apple key active. Bit 5 = updated modifier latch without keypress. Bit 4 = keypad key active. Bit 3 = repeat active. Bit 2 = caps lock active. Bit 1 = control key active. Bit 0 = shift key active

Hail and well met, Eamon adventurer.

COMPUTIST has news of great import for loyal supporters of Eamon and members in good standing at the Main Hall. There is a newsletter for Eamon Adventure Buffs.

"The Eamon Adventurer's Guild"

Tom Zuchowski is the editor and driving force behind it. The newsletter is printed on 8 1/2" x 11" bond, double sided with dot matrix type. It is non-profit (it's his hobby) and Tom's avowed purpose is to maintain a single point collection and clearing house for Eamon Adventures. He also intends to fix all known bugs in these adventures.

This is a grand concept worthy of support. We highly recommend that you subscribe. This newsletter will help you keep in touch with other Eamon adventurers. But more important, it will improve the state of Eamon adventures and encourage new adventures to be written. Tom has accomplished a great deal along these lines already, not only fixing bugs but also improving the Eamon Main program and authoring the version 7.0 Dungeon Designer Disk.

The "Guild" is printed quarterly. A 1 year subscription is:
US & Canada: \$7.00 Foreign: \$12.00 (U.S. funds)

The Eamon Adventurer's Guild
7625 Hawkhaven Dr.
Clemmons, NC 27012
(919) 766-7490

Yes! I want to support Eamon adventure and encourage new adventures to be written. Sign me up for a one year subscription.

- US & Canada: \$7.00
 Foreign: \$12.00 (U.S. funds)

I also want to order some back issues at \$2 each:

- Jun'88 Sep'88 Dec'88
 Mar'89 Jun'89 Sep'89
 Dec'89 Mar'90 Jun'90
 Sep'90 Dec'90 Mar'91

Total enclosed _____

Name _____

Address _____

City _____ State _____ Zip _____

Eamon Adventure for only \$1 (or less) each

Adventure Gaming doesn't have to cost a lot. The Eamon Adventure Gaming system was created by Donald Brown and placed into the public domain. Since then it has been updated and improved by game players all over the world. Take a look at what \$1 will buy. (Get free games too.) *Note: Some Adventures are multi-part and take more than one disk. Be sure you have selected all of the disks.*

- | | | | | |
|---|---|--|--|--|
| <input type="checkbox"/> 1 Main Hall & Beginners Cave | <input type="checkbox"/> 42 Alternate Beginners Cave | <input type="checkbox"/> 83 The Twin Castles | <input type="checkbox"/> 122 The Valley of Death | <input type="checkbox"/> 157 Pathetic Hideout of Mr R. |
| <input type="checkbox"/> 2 The Lair of the Minotaur | <input type="checkbox"/> 43 Priests of Xim! | <input type="checkbox"/> 84 Castle of Riveneta | <input type="checkbox"/> 123 Wizard of the Spheres | <input type="checkbox"/> 158 The Lair of Mr Ed |
| <input type="checkbox"/> 3 The Cave of the Mind | <input type="checkbox"/> 44 Escape from the Orc Lair | <input type="checkbox"/> 85 The Time Portal | <input type="checkbox"/> 124 Assault on Dolni Keep | <input type="checkbox"/> 159 The Bridge of Catzad-Dum |
| <input type="checkbox"/> 4 The Zyphur Riverventure | <input type="checkbox"/> 45 SwordQuest | <input type="checkbox"/> 86 Castle Mantru | <input type="checkbox"/> 125 The Mattimoe Palace | <input type="checkbox"/> 160 Monty Python & Holy Grail |
| <input type="checkbox"/> 5 Castle of Doom | <input type="checkbox"/> 46 Liferequest | <input type="checkbox"/> 87 Caves of Hollow Mountain | <input type="checkbox"/> 126 The Pyramid of Anharos | <input type="checkbox"/> 161A Operation Endgame |
| <input type="checkbox"/> 6 The Death Star | <input type="checkbox"/> 47 FutureQuest | <input type="checkbox"/> 88 The Shopping Mall | <input type="checkbox"/> 127 The Hunt for the Ring | <input type="checkbox"/> 161B Operation Endgame |
| <input type="checkbox"/> 7 The Devil's Tomb | <input type="checkbox"/> 48 Picnic in Paradise | <input type="checkbox"/> 89 Super Fortress of Lin Wang | <input type="checkbox"/> 128 Quest of Erebor | <input type="checkbox"/> 161C Operation Endgame |
| <input type="checkbox"/> 8 The Abductor's Quarters | <input type="checkbox"/> 49 The Castle Kophinos | <input type="checkbox"/> 90 The Doomsday Clock | <input type="checkbox"/> 129A Return to Moria | <input type="checkbox"/> 162 Eamon 7.0 Demo Adventure |
| <input type="checkbox"/> 9 Assault on the Clonemaster | <input type="checkbox"/> 50 Behind the Sealed Door | <input type="checkbox"/> 91 FutureQuest II | <input type="checkbox"/> 129B Return to Moria | <input type="checkbox"/> 163 The Sands of Mars |
| <input type="checkbox"/> 10 The Magic Kingdom | <input type="checkbox"/> 51 The Caves of Eamon Bluff | <input type="checkbox"/> 92 The Fugitive | <input type="checkbox"/> 130 Haradwaith | <input type="checkbox"/> 164 A Real Cliffhanger |
| <input type="checkbox"/> 11 The Tomb of Molinar | <input type="checkbox"/> 52 The Devil's Dungeon | <input type="checkbox"/> 93 Flying Circus | <input type="checkbox"/> 131 Nucleus of the Ruby | <input type="checkbox"/> 165A Animal Farm |
| <input type="checkbox"/> 12 The Quest for Trezore | <input type="checkbox"/> 53 Feast of Carroll | <input type="checkbox"/> 94 Blood Feud | <input type="checkbox"/> 132 Rhadshur Warrior | <input type="checkbox"/> 165B Animal Farm |
| <input type="checkbox"/> 13 Caves of Treasure Island | <input type="checkbox"/> 54 Crystal Mountain | <input type="checkbox"/> 95 The Maze of Quasequeton | <input type="checkbox"/> 133 The Final Frontier | <input type="checkbox"/> 166A Storm Breaker |
| <input type="checkbox"/> 14 Furioso | <input type="checkbox"/> 55 The Master's Dungeon | <input type="checkbox"/> 96 The Chamber of the Dragons | <input type="checkbox"/> 134 Pyramid of the Ancients | <input type="checkbox"/> 166B Storm Breaker |
| <input type="checkbox"/> 15 Heroes Castle | <input type="checkbox"/> 56 The Lost Adventure | <input type="checkbox"/> 97 The House of Secrets | <input type="checkbox"/> 135 The Tomb of Evron | <input type="checkbox"/> 166C Storm Breaker |
| <input type="checkbox"/> 16 The Caves of Mondamen | <input type="checkbox"/> 57 The Manxome Foe | <input type="checkbox"/> 98 Slave Pits of Kzorland | <input type="checkbox"/> 136 The Mountain Fortress | <input type="checkbox"/> 167 Expedition to the Darkwoods |
| <input type="checkbox"/> 17 Merlin's Castle | <input type="checkbox"/> 58 The Land of Death | <input type="checkbox"/> 99 In the Clutches of Torrik | <input type="checkbox"/> 137 The Ruins of Ivory Castle | <input type="checkbox"/> 168 The High School of Horrors |
| <input type="checkbox"/> 18 Hogarth Castle | <input type="checkbox"/> 59 Jungles of Vietnam | <input type="checkbox"/> 100 Sorcerer's Spire | <input type="checkbox"/> 138 Starfire | <input type="checkbox"/> 169 The Black Phoenix |
| <input type="checkbox"/> 19 Death Trap | <input type="checkbox"/> 60 The Sewers of Chicago | <input type="checkbox"/> 101 Ground Zero | <input type="checkbox"/> 139 Peg's Place | <input type="checkbox"/> 170 Ragnarok Revisited |
| <input type="checkbox"/> 20 The Black Death | <input type="checkbox"/> 61 The Harpy Cloud | <input type="checkbox"/> 102 The Eamon Railroad | <input type="checkbox"/> 140 Beginner's Forest | <input type="checkbox"/> 171 The Pyramid of Cheops |
| <input type="checkbox"/> 21 The Quest for Marron | <input type="checkbox"/> 62 The Caverns of Doom | <input type="checkbox"/> 103 Top Secret | <input type="checkbox"/> 141 The Infested Fortress | <input type="checkbox"/> 172 The Mountain of the Master |
| <input type="checkbox"/> 22 The Senator's Chambers | <input type="checkbox"/> 63 Valkenburg Castle | <input type="checkbox"/> 104 The Lost World | <input type="checkbox"/> 142 The Beermeister's Brewery | <input type="checkbox"/> 173 The House that Jack Built |
| <input type="checkbox"/> 23 The Temple of Ngurct | <input type="checkbox"/> 64 Modern Problems | <input type="checkbox"/> 105 The Strange Resort | <input type="checkbox"/> 143 The Alternate Zone | <input type="checkbox"/> 174 Escape from Granite Hall |
| <input type="checkbox"/> 24 Black Mountain | <input type="checkbox"/> 65 The School of Death | <input type="checkbox"/> 106 Camp Eamon | <input type="checkbox"/> 144 Gartin Manor | <input type="checkbox"/> 175 Anatomy of the Body |
| <input type="checkbox"/> 25 Nuclear Nightmare | <input type="checkbox"/> 66 Dungeons of Xenon | <input type="checkbox"/> 107 The Last Dragon | <input type="checkbox"/> 145A Buccaneer! | <input type="checkbox"/> 176 Dirtie Trix's Mad Maze |
| <input type="checkbox"/> 26 Assault on the Mole Man | <input type="checkbox"/> 67 Chaosium Caves | <input type="checkbox"/> 108 The Mines of Moria | <input type="checkbox"/> 145B Buccaneer! | <input type="checkbox"/> 177 Shippe of Fools |
| <input type="checkbox"/> 27 Revenge of the Mole Man | <input type="checkbox"/> 68 The Smith's Stronghold | <input type="checkbox"/> 109 The Forest of Fear | <input type="checkbox"/> 146 The House of Horrors | <input type="checkbox"/> 178 The Alien Intruder |
| <input type="checkbox"/> 28 The Tower of London | <input type="checkbox"/> 69 The Black Castle of NaGog | <input type="checkbox"/> 110 Fire Island | <input type="checkbox"/> 147A The Dark Brotherhood | <input type="checkbox"/> 179 The Wizard's Tower |
| <input type="checkbox"/> 29 The Lost Island of Apple | <input type="checkbox"/> 70 The Tomb of Y'Golomac | <input type="checkbox"/> 111 A Vacation in Europe | <input type="checkbox"/> 147B The Dark Brotherhood | <input type="checkbox"/> 180 Gamma 1 |
| <input type="checkbox"/> 30 The Underground City | <input type="checkbox"/> 71 Operation Crab Key | <input type="checkbox"/> 112 Hills of History | <input type="checkbox"/> 148 Journey to Jotunheim | <input type="checkbox"/> 181 The Eamon Sewer System |
| <input type="checkbox"/> 31 The Gauntlet | <input type="checkbox"/> 72 House on Eamon Ridge | <input type="checkbox"/> 113 The Life-Orb of Mevtrelek | <input type="checkbox"/> 149A Elemental Apocalypse | <input type="checkbox"/> 182 Farmer Brown's Woods |
| <input type="checkbox"/> 32 House of Ill Repute | <input type="checkbox"/> 73 The Deep Canyon | <input type="checkbox"/> 114 Thror's Ring | <input type="checkbox"/> 149B Elemental Apocalypse | <input type="checkbox"/> 183 The Boy and the Bard |
| <input type="checkbox"/> 33 The Orb of Polaris | <input type="checkbox"/> 74 DharmaQuest | <input type="checkbox"/> 115 The Ring of Doom | <input type="checkbox"/> 149C Elemental Apocalypse | <input type="checkbox"/> 184 Quest for Orion |
| <input type="checkbox"/> 34 Death's Gateway | <input type="checkbox"/> 75 Temple of the Guild | <input type="checkbox"/> 116 The Iron Prison | <input type="checkbox"/> 149D Elemental Apocalypse | <input type="checkbox"/> 185 The Body Revisited |
| <input type="checkbox"/> 35 The Lair of Mutants | <input type="checkbox"/> 76 The Search for Yourself | <input type="checkbox"/> 117 Dungeon of Doom (40 col) | <input type="checkbox"/> 150 Walled City of Darkness | <input type="checkbox"/> 186 Beginners Cave II |
| <input type="checkbox"/> 36 The Citadel of Blood | <input type="checkbox"/> 77 Temple of the Trolls | <input type="checkbox"/> 117 Dungeon of Doom (80 col) | <input type="checkbox"/> 151 EamonS.A.R.-1(Deneb Raid) | <input type="checkbox"/> 187 Batman! |
| <input type="checkbox"/> 37 Quest for the Holy Grail | <input type="checkbox"/> 78 The Prince's Tavern | <input type="checkbox"/> 118 Pitfall | <input type="checkbox"/> 152 The Computer Club of Fear | <input type="checkbox"/> 188 Encounter: The Bookworm |
| <input type="checkbox"/> 38 City in the Clouds | <input type="checkbox"/> 79 The Castle of Count Fuey | <input type="checkbox"/> 119A Grunewalde | <input type="checkbox"/> 153 Lost! | <input type="checkbox"/> 189 The Ruins of Belfast |
| <input type="checkbox"/> 39 Museum of Unnatural History | <input type="checkbox"/> 80 The Search for the Key | <input type="checkbox"/> 119B Grunewalde | <input type="checkbox"/> 154 A Trip to Fort Scott | <input type="checkbox"/> 190 Shift Change at Grimmwax |
| <input type="checkbox"/> 40 Daemon's Playground | <input type="checkbox"/> 81 The Rescue Mission | <input type="checkbox"/> 120 Orb of My Life | <input type="checkbox"/> 155 Tomb of the Vampire | <input type="checkbox"/> 191 Enhanced Beginners's Cave |
| <input type="checkbox"/> 41 Caverns of Lanst | <input type="checkbox"/> 82 Escape from Mansi Island | <input type="checkbox"/> 121 Wrenhold's Secret Vigil | <input type="checkbox"/> 156 The Lake | <input type="checkbox"/> 192 Mean Streets |

Send me the Complete set of Eamon (all adventures & utilities) for: \$125.00

Total number of Adventure disks _____ x \$1 each = _____

If total # of disks ordered is less than 10, add \$4 for postage & handling. _____

Washington State residents add 7.8% sales tax. _____

Name _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

Visa _____ Exp _____

Signature _____

COMPUTIST, 33821 Orville Rd. E, Eatonville WA 98328-9590 (206) 832-3055

Free

Adventures

Use the total number of adventures ordered to determine how many free adventures you get.

Be sure and check the boxes of your free disks that you want but **do not** include free disks when figuring total number of disks ordered.

# of disks at \$1	# of Free disks
1-9	0
10-19	2
20-29	5
30-39	9
40-49	14
50-59	20
60-69	27
70-79	35
80-89	44
90-99	54
100-109	65
110-119	77
120-129	90

Complete set of Eamon

All 225 disks (includes all adventures plus designer and utility disks.)\$125

- 193 The Creature of Rhyll
- 194 Attack of the Kretons
- 195 The Training Grounds
- 196 The House of Horrors
- 197 Star Wars - Tempest One
- 198 Revenge of the Bookworm
- 199 Quest of the Crystal Wand
- 200 The Lost Isle
- 201 The Caverns of Vanavara
- 202 The Plain of Srevi
- 203 Lotto's Masterpiece
- 204A Sanctuary
- 204B Sanctuary
- 205 Utterly Outrageous
- 206 Curse of the Hellsblade

- Dungeon Designer Diskette v7.0
- Multi-Disk Supplement (DDD7.0)
- Dungeon Designer Diskette v6.2
- Eamon Utilities Diskette
- Graphics Main Hall

matically load the rest of the program. This edit totally bypasses the asking of the CP question.

Terry Waskowich &
Jay Kitaguchi

Softkey for...

Windwalker IIe

Origin

Requirements:

2 Blank Disks
Copy II Plus
Sector Editor
COPYA

Windwalker is a really neat game. The main idea behind this game is to become a Windwalker yourself, and advancing your stats to a point where Moebius will grant you the new status. But before Moebius can grant you the new status you have to answer a question. The following will tell you how to backup Windwalker Magnetic Scroll Side 1-4 and take out the question.

Step-by-step

1. RUN COPYA and when it has finished loading hit ctrl-reset.

2. At the Applesoft prompt (I) type in:
70 REM

CALL-151

B96D:BD 8C C0

B970:10 FB 2A 85 26 BD 8C C0

B978:10 FB 25 26 99 2C 00 88

B980:10 EB BD 8C C0 10 FB C9

B988:DE F0 0F EA BD 8C C0 C9

B990:AB D0 0B 38 A5 2D E9 11

B998:85 2D 18 60 EA EA 18 60

3D0G

RUN

Copy Windwalker's Magnetic Scroll

1. Use COPYA like you would normally. After you have done that, make a backup of Windwalker's Magnetic Scroll 2 to 4 with any copier or fast disk copier.

3. At this point, put your original disks away. Edit the copies of Magnetic Scroll 1 to 4.

Disk #1

Trk	Scr	Byte	From	To
01	04	\$89	F0 0E	F0 0F

Disk #2

Trk	Scr	Byte	From	To
1D	05	\$0E	20 38 43 D0 0D	A9 01 18 90 0D

Disk #4

Trk	Scr	Byte	From	To
07	00	\$0E	20 38 43 D0 0D	A9 01 18 90 0D

After you have written those patches to the correct disks, you will then have a complete backup of your original and Moebius will not ask the question before he enhances your character.

Softkey for...

Windwalker GS

Origin

Requirements:

Blank Disk (3.5")

Fastcopier (3.5")

Blockwarden/Block Editor

Here is a quick way of taking the word protection off this game. The word protection will come up when Moebius is ready to enhance your character during the game.

Step-by-step

1. Make a Copy of Windwalker GS Disk A with any disk copier.

2. Use Block Warden or your favorite block editor to edit the copy. (Block Warden [F]ollow /WINDWALKER .A/WIND.SYS16 on REL BLOCK 97.)

Blk	Byte	From	To
\$190	\$0C7	22 84 28 03	AF 84 28 03
	\$0D9	22 B1 25 03	AF B1 25 06
	\$103	22 12 1E 03	AF 12 1E 03
	\$114	22 E3 18 03	AF E3 18 03
	\$11A	22 8F DA 00	AF 8F DA 00
	\$11E	22 79 26 03	AF 79 26 03
	\$122	22 79 26 03	AF 79 26 03
	\$17F	F0 02 80 23	EA EA 80 23

3. Write the block back to the copy.

Now the word protection won't even show up and Moebius will automatically enhance your character.

Brian A. Troha &
Safaa Abdulla &
Terry Waskowich

Softkey for...

Pipe Dreams

Lucasfilm

Requirements:

512K Apple IIgs or 128K Apple II series

Disk copier (5.25" or 3.5")

Disk editor (5.25" or 3.5")

The program package comes with both the GS and IIe versions in the same box. The game is very good, however before you can play the first game you will have to look up something from a code-wheel. I didn't like the idea of having to use the code-wheel so we decided to take it out. After following the code with a couple of friends, we found how to bypass the copy protection completely. By searching for C9 8D (CMP#\$8D - the value of the return key, for the IIe version also try searching for AD 00 C0 or LDA \$C000, which is the keyboard location) we found the routine, tracing forwards and backwards we found a simple call to the CP. Changing the JSR to LDA gets rid of the entire requestor screen.

After playing the game for awhile a "demo" screen came up and said something about "We here at LucasFilms hope you are enjoying.." So, LucasFilms has put some kind of flag in the program and if not set it will go to the above described screen. So we searched for references to the text and uncovered the routine responsible for checking the flag. The routine checks the high bit of a location on the zero page then checks a location for a value and if it's a certain value or higher then goes to the "demo" screen. Tracing it back to the original JSR and changing it to a LDA results in a totally deprotected copy of Pipedreams. The same information applied to both the GS and IIe versions so cracking the IIe version went very fast. With the edits installed, you will never see any signs of the requestor screens. Follow these instructions:

Step-by-step

1. Make a copy of the game disks (both the GS & IIe versions).

2. Make the following edits by computer version:

512K Apple IIgs

Blk	Byte	From	To
\$19D	\$1F6	20 C6 47	AD C6 47
\$19E	\$C5	20 A8 37	AD A8 37

128K Apple II series

(ProDOS block \$62)

Trk	Scr	Byte	From	To
\$0C	\$0B	\$1C	20 89 B2	AD 89 B2
		\$D4	20 60 A4	AD 60 A4

3. Write the edits back to the copy.

Safaa Abdulla &
Brian A. Troha

Softkey for...

Hostage

Mindscape Inc.

Requirements:

512K Apple IIgs

3.5" disk copier

3.5" Disk editor

Recently, I purchased the game Hostage from Mindscape, Inc., and although the gameplay is excellent, the off-disk copy protection (documentation check) is far from it. So I decided that I would have to do something about it.

When I booted the disk, it showed the good ole' ProDOS 16 loader, then it went to a SHR picture of the Mindscape logo. When disk access stopped, a window opened up in the middle of the screen asking to "Please, enter word X on line Y of page Z." "Hmmm...", I said (Yes, I actually said "Hmmm...") as I grabbed a piece of paper to scribble down the text "Please, enter". Next, I copied the program onto another disk (NEVER work with your originals). Then I booted up my good friend Block Warden and let him search for the text on the disk. Sure enough, there was the text on block \$019A at byte \$000B. In hex, it was: \$50 6C 65 61 73 65 2C 20... I figured that this was enough to find the text in memory...

Next, I booted the new disk. This time, when it got to the protection, I just entered garbage...It returned with a loud >BOOM< (Oops...I had booted Modulae earlier...sorry Mom!) After I turned off the stereo, I entered more garbage. After the third attempt, the program bailed into the "Start Next Program" selector. Unfortunately, the control panel was locked out...I hit CONTROL-RESET, and it went to the \$0602 UNABLE TO RESET screen with the bouncing apple. I hit reset again and voila! I was in the monitor! Using the IIgs built-in pattern search commands, I was able to locate the text in bank \$02 at byte \$02A6.

\50 6C 65 61 73 65 2C 20\<2/0.FFFFFF

02/02A6:

Next, I searched for references to \$02A6, hoping it would be that easy...

\A6 02\<02/0.FFFFFF

02/541E:

02/8BD3:

02/8E6A:

Great! There were my references... After further examination, the second and third references were not what I was looking for, so I proceeded to examine the code around \$541E to see what it had to offer. The code at \$541A looked something like this:

02/541A:	PEA 0002	The bank where the text is
	PEA 02A6	The address of the text
	LDA 0BD6	
	PHA	Push some other stuff on the stack
	LDA 0BCA	
	PHA	More stuff pushed
	...	and it continues

It looks like, to me, that it's pushing the address of the text onto the stack, and

setting up some other parameters for calling the windowing routine. Backtracking through the code, I found a RTL (\$6B) at byte \$53F4, so taking a guess, I figured that the routine started at \$02/53F5. Doing another pattern search:

\F5 53\<1/0.FFFFFF

01/B740:

Here was the reference (hopefully) to the routine that I wanted...

01/B73F: JSL 0253F5 Jump to our windowing routine!

JSL 029F95 Hmmm... Could this be related? We'll see later.

Looking at the code before our jump, I found some branches that would not be changed by ProDOS 16 or GS/OS upon load time (\$F4 01 00 AE) and searched for that on the disk. I found it on block \$01B6, and saw afterwards at byte \$0057 a \$22 (JSL). I changed this to a \$AF (LDAL) and wrote it back out to disk. I booted the disk again, and this time, after the Mindscape page, I saw our "Start Next Program" requester. "Hmmm..." (Yes, I said it again) After I saw this, I checked out my good old IIgs reference books, and found this to be call \$0029 from the ProDOS 16 MLI. SOOO...From here, I hit reset again (twice...) and got back to the monitor to search for a \$22 A8 00 E1 29 00, which properly formatted is:

\22 A8 00 E1 29 00\<2/0.FFFFFF

02/9FBA:

02/9FBA: JSL E100A8

0029

000206B9 ;we won't worry with this stuff...

And, if you remember, this is right after the SECOND jump that we found before... My first guess was to take this jump out all together. So, remembering that the jump was \$04 bytes after the first jump we edited on disk, I blindly changed the \$22 at byte \$005B to an \$AF.. WRONG! Well, the program DID load, but '?????'s were scrolling on the screen, as well as other funky stuff...so I promptly changed this back...So, I went back to the MLI call that we had earlier... The code looked something like this:

	CMP #0001	
	BEQ 9FC6 (+0C)	
02/9F95	JSL E100A8	The call to the MLI
	0029	The code for the "Start
	000206B9	Next Program"
	BRK 0000	
	PEA 0002	This is where the BEQ
		above branches too...
	PEA 3D73	
	...	and the code continues...

So, can you see where the check is made to see if the word was entered correctly? Good. So let's try changing the BEQ 9FC6 to a BRA 9FC6 (Block \$01ED, byte \$016F, from \$F0 to \$80) and see what happens. It works. You now have a fully unprotected copy of Hostage for the IIgs!

These edits change the JSL 5/1241 to LDA 5/1241, in effect killing the jump to the manual check. Then we change the BEQ (Branch on Equal) to BRA (BRanch Always) to force the program to continue no matter what value the CP flag holds

Step-by-step

1. Make a copy of the Hostage game disk.

2. Make the follow edits to the copy only:

Blk	Byte	From	To
\$1B6	\$57	22 41 12 05	AF 41 12 05
\$1ED	\$16F	F0 0C	80 0C

3. Write the edits back to the copy.

Softkey for...

Mickey's Crossword Puzzle Maker Disney

Requirements:

64K Apple II series
3.5" disk copier
3.5" disk editor

Like the name implies, Mickey's Crossword Puzzle Maker will make and play crossword puzzles. This game is designed for younger children and as such demands an unprotected backup. The copy protection was in the form of a word protection where you had to look up a crossword puzzle in the back and give the answer to X up or Y across. This is very annoying for adults, let alone children so we decided to remove the entire routine. After a little code snooping we found an edit that would totally skip the requestor screen. Then after trying to boot the disk on different computers we found a routine that seemed to cause crashes on some machines. We tracked it down and took out the call to so the program would run on all machines.

Step-by-step

1. Make a copy of the program disk.
 2. Make the following edits to the disk.
- | Blk | Byte | From | To |
|------|-------|----------|----------|
| \$2A | \$119 | 20 1D 63 | EAEAEA |
| \$2F | \$08 | 20 75 E0 | 18 90 4C |
3. Write the edit back to the disk.
 4. Give the newly deprotected copy to your kids to use!

Kilroy OR

Advanced Playing Technique for...

Space Rogue

Origin

Space Rogue is a new game from Origin (see Product Monitor, issue #74) that looks a little like Autoduel. You have to build up your spaceship, follow the clues and unravel the evil plot, and then save the galaxy. No problem! That is if you don't mind a SLOW pace at the beginning while you earn the money needed to build up your spaceship.

All of the attributes are stored in the SAVE file of the side B (/DATA) disk. The following edits are all in track 20, sector 04 unless otherwise noted:

On track 20, sector 04

- 0E-19 Name: end with 00 (12 characters max)
- 1E-20 Date: Day/Month(1-12)/Year
- 1A-1C Location: Quadrant/Sector (00,00 is the lower LH corner and 31,00 is the lower LH corner). Quadrants:
- 00 Karonus
 - 01 Deneb
 - 02 Gryphon
 - 03 Arcturus
 - 04 Nar'see
 - 05 Bassruti
 - 06 Sigure
 - 07 Zed N27
 - 09 Ja-Kam
- 23-24 Bounty: hex backwards
- 25-26 Money: hex backwards (57 0E = 3671)
- 27 Repute: values are:
- 00 unknown

- 01 pesky
- 02 fierce
- 03 deadly
- 04 ace
- 05 INVINCIBLE

- 2B Status: Pirates
- 2C Status: Guild
- 2D Status: Imperium Values are:
- 00 neutral
 - 01 liked
 - FF disliked
 - FE hated
- 2F Missiles: # of SM-1 missiles
- 30 Missiles: # of NOVA missiles
- 31 Missiles: # of PLASMA missiles
- 32-33 Armor: hex backwards (E8 03 = 1000) *Armor Remaining 20 08
- 1A 1B hex backwards
- 34 ECM: value 4B = 75% ECM
- 35 Installed: a value of 3F "installs" the ECM & shield options
- 36 Laser: types are:
- 01 garnet
 - 02 beryl
 - 03 sapphire
 - 04 particle beam (This is the one you want!)
- 37 Shields: fore/aft, a value of 33 = Fore 3/Aft 3
- 74 Cargo pods: 08 max

Item List (01 unless otherwise noted)

- 3A Stealth Box
- 3B Beam Lock
- 3C Turbo Thruster
- 3D Repair Droid
- 3E Null Damper
- 3F Key Card:
 - 01 fits locks at: Hiathra Cantina; Micon I; & ?
 - 02 ??
 - 03 fits locks at: Bassruti, Gut's office; Koth Carrier; Zed N27 storeroom (crew's quarters); Con-Vec East, Omas Tyran's room
- 40 Manchi Egg
- 41 CRC-07 Form
- 42 Transmutation Coil
- 43 NS Booster
- 44 Statuette
- 45 Malir Artifact
- 46 Ruby Cube
- 47 Tiwa's Letter
- 48 Psionic Shield
- 49 Microchip
- 4A Pilot's License
- 4B Forged Cargo Papers
- 4C Amoebic Lenses
- 4D Blog Globe
- 4E Dilithium Gem

Playing Tip for...

Space Rogue

Origin

Look for messages while "charting".

Make your first stop at the Micon I mining station and see Sir Eld. Your second stop should then be at the Hiathra station and the Imperial representative Orellian. You'll need to pass your pilot's exam: 3.26, type O, F=ma, and Imperium. As long as you're there, ask for a CRC-07 form. You'll need it later.

Search outside the Ross Mining Station.

Search the storeroom in the Free Guild outpost, but watch out for an alarm plate on the right side.



William Degelmann CO

Bitkey for...

Black Magic

?

Requirements:

Copy II+ 9.0

Use the Parm Bit-Copy.

Bitkey for...

Ultima V

?

Requirements:

Copiers: LS 6.0 and Copy II 9.0

For the Boot Side: Use LS 6.0, Bitcopy, and copy tracks 11 to 22. Then use Copy II 9.0 to copy tracks 00 to 11.

For the Towne side: Use Copy II 9.0 to copy the disk using 'Disk w/Format'.

For all other disks: use Ultima V copy under Normal Bitcopy.

Bitkey for...

Realms of Darkness

?

Requirements:

Echo 1.0, Copy II+ 9.0

First; For the Boot, use Echo 1.0 and press return to default all the values correctly. You will see errors on the disk, but these are empty tracks. When the copy is done, recopy track 08 over and over until it copies.

Second; With your partially created Boot disk, load it in the drive, and use the "Repair Disks" option to see if you have any errors. If you do, the program will fix them.

Third; For the Scenario disks, use the Copy II+ 9.0 sector copy on both sides of the disk. When you are done, again; check it on the Boot disk for any errors.

Bitkey for...

Saracen Gunslinger Guild of Thieves Police Quest Silicon Dreams

?

Requirements:

Echo 1.0 bit copier

For all these Programs: Press return until you see 'Insert Disks' and press return to start copying.

Bitkey for...

Wizardry Series Sir Tech

Requirements:

Copiers: Essential Data Dup. II and Copy II+

For the Scenario side: select copy a disk, and run down the menu, until you can choose a number of copying choices and press #5. Use essential dup.

Warning! Make sure you insert the main original disk in drive 2, and the copy in drive 1.

For the Boot side: find the program under the parm entries on the disk, and copy with a Bitcopy. Use Copy II.

Warning! Make sure you put a write-protect tab on the boot side, or it will not run!

Bitkey for...

Xyphus

?

Requirements:

Locksmith 5.0

For the Boot side; use Bitcopy, and press these:

SYNCRONIZE: NO
TRACK LENG: M
ERROR RETR: YES

Copy Tracks 00 to 06, and 08, 10, 11

For tracks 07 and 09 on the Boot disk, use the Bitcopy again, and press return until it says to insert the disks.

For the Back side: use the Bitcopy, and again, press return until it says to insert the disks.

Bitkey for...

Sierra Boot Disks

Sierra OnLine

Requirements:

Copy II+ 9.0

Using the Bit-Copier, look for the parm: "Leisure Suit Larry (Alt 1)", and load it up. This will copy these Games:

Man Hunter: New York
Leisure Suit Larry
Police Quest
King's Quest Series
Space Quest Series

Who knows what else it will copy from Sierra? I'll find out soon.

If you would like a bitkey done for you, contact me through the Computist RDEX editor. I will be glad to help you find a solution quickly.

Ⓢ Does anyone have an MODEM that they are no longer using and would like to sell? I am looking to buy a 1200 or 2400 baud MODEM for under \$100. I prefer an internal (for a IIe) but I'll take an external if it comes with a serial card. Again, I can be reached thru the RDEX editor.

Carl D. Purdy IA

In reading the most wanted list I saw a program that I deprotected some time back.

Rear Guard

Adventure International

Requirements:

- COPYA
- Sector Editor such as CopyII+ or Diskedit (I prefer Diskedit)
- Disk Searcher such as CopyII+ or Core Disk Searcher
- Blank disk

1. First copy the disk using a modified version of COPYA that will ignore errors. When the copy program gets to the last track it will complain a lot, be patient, it will finish it's task.

RUN COPYA

```
ctrl reset          at the menu
POKE 929,24
POKE 47426,24
RUN 80
```

2. When the copy program is finished insert your disk searcher and search for the sequence BD 89 C0. You should find one occurrence on track 0, sector 7 and another on another track. Don't change the one on track 0. I found the bytes on track 03, sector 03, byte 03. I changed the sequence, using a sector editor, to 18 60 EA.

Trk	Sct	Byte	From	To
\$03	\$03	\$03-05	BD 89 C0	18 60 EA

3. Write track 3 or whatever track you find the routine on back to disk.

You now have a deprotected copy of Rear Guard. By the way, the deprotected copy will not growl the disk drive just before starting the game as the original did.

☺ I am interested to know if anyone has deprotected Where in the USA is Carmen Sandiego. I have seen Bitkeys for Carmen USA but never a Softkey. (This is the DOS 3.3 version not the PRODOS version).

Krakovicz

The Basics of Cracking (part 7)

Beyond DEMUFFIN: non-standard disk encoding techniques and diskbit tidbits

Recently, we have discussed the unprotection of disks with a modified DOS, primarily through the use of DEMUFFIN PLUS. This time we'll talk a little about some other approaches to converting modified DOS disks, then get into some slightly heavier stuff about those protection techniques which go way beyond modifying DOS.

In addition to DEMUFFIN PLUS, two programs have recently become available for unprotecting a disk with modified DOS: COPYB and ADVANCED DEMUFFIN. Since, as we discussed earlier, most mods are made to the RWTS portions of DOS, all that's really necessary to remove the primary protection is to read the disk into memory using its own RWTS, then switch in a normal RWTS and write it back out to another disk. Since the file manager is not invoked as it is in DEMUFFIN PLUS, this approach has the added advantage that a disk with fairly normal sector structure but no DOS file structure or catalog can still be converted. In the early days, the technique was this: store both RWTS's in memory, then use the Inspector to read in about 8 tracks. Move the normal RWTS in with the monitor,

then write out the tracks to an initialized disk. About 5 times around makes a COPYA disk and Jack a dull boy. Fortunately, things have improved.

COPYB is a modification of COPYA which automatically swaps the RWTS routines for you. To run it, you boot the protected program and interrupt it, then move the RWTS routines from \$B700 to \$8000 (you can also use the read-in technique described in Basics 105 (File #5) to put the modified RWTS into memory, and you can keep a library of RWTS's if you find people like Muse and SSI using a particular one over and over. These can simply be loaded as Bfiles into \$8000 after booting COPYB). Booting COPYB and answering one cryptic and four fiddish questions allows you to produce a COPYA disk, including automatic initialization of the target disk. Reasonably complete instructions, written by the author who has achieved national prominence for verbal diarrhea, accompany the program, so we won't belabor them here.

A much more complete program called Advanced Demuffin has recently issued from Corrupt Computing, under the able authorship of "the STACK" and "the Inspector". It also makes unprotected copies via RWTS swaps, but is much more user-friendly and versatile. Thorough softdoc accompanies that program as well, so we needn't dwell on it, except to congratulate the authors on an excellent and highly professional contribution to the art of unprotection.

Except for some clever and well-hidden secondary protection, there is not much that a protector can do these days with a modified DOS that we can't undo in short order with the tools and techniques available to us. Why are there still some programs that take a long time to Krack? More extensive modifications, extending even to complete custom DOS's. Here, however, we start to separate the men from the boys, since writing your own operating system, no matter how limited, costs money. While DOS modifications, even with several variations, can be whipped out in a few minutes by any klutzy hacker, reading and writing in ways not sanctioned by the gods of Apple DOS require hiring someone who knows his stuff, and he also usually knows how much he's worth. This has advantages for us as well, since publishers will try to get their money's worth out of an expensive system by using it on as many products as possible. Once broken, the principles can be readily applied to all disks of the same generation of protection.

If a publisher is going to go beyond modifying DOS, he will normally also abandon standard track and sector format for something which affords greater security and ease of use (sometimes, since the Apple disk hardware is so flexible, formats which were born on entirely different systems find their way into Apple protection schemes). Games, especially, have much simpler structure, and are readily adapted to a format with less complexity. Since space on a game disk is usually not at a premium, a very common simplification is to eliminate sectoring altogether, and make each track one big sector. This not only simplifies the program that has to read the disk, but can also dramatically increase the data transfer rate (Sirius's Hadron, brought in a full 48K in just over four seconds—Eat your heart out, DOS). Before we discuss some of the formats

used, we have to take a much closer look at the way information is actually read from a disk.

There are a few absolute laws of disk writing and reading which must be observed, and several minor statutes which may be violated with only a summons. The real, deep down, true way that data is recorded on any disk is by way of "magnetic flux changes", that is, reversals in the direction of magnetization of a thin coating of iron oxide on the disk surface. We all recall fondly the science experiments with iron filings and a bar magnet; disk recording technology is based on making the particles very small, and immobilizing them on the disk so they can be examined later for the state of their magnetization. Diskette reading is actually a (gasp!) analog process, and is made digital by some clever circuitry just downstream of the read head. This circuitry senses the magnetic field over a precisely defined time interval, and translates a *change* (reversal) in the direction of magnetization to a digital "one", and interprets *no change*, or the absence of reversal, as a "zero".

On a 5.25" diskette, the changes are assessed every 4 microseconds (usec), and the digital "bits" read are used to build up a disk byte or "nibble" (in case you were curious, the disk spins at 300 rpm which is 5 revolutions per second, or 200 milliseconds (msec) per rotation. Since 8 bits = one byte, a byte is read every 32 usec, or 0.032 msec, and each revolution of the disk corresponds to 200 msec/0.032 msec or about 6000 (decimal) bytes. This is roughly \$1800 bytes per track, which is about the number of bytes you normally see displayed during a nibble count with Nibbles Away or Locksmith).

It's not too bad a physical picture to represent the orientation of the magnetic fields with arrows (up and down arrows would be nicer, but the Apple screen no gots). In figure 9 below, the orientation of magnetic "domains" on the disk for 9 bits are represented by arrows

Notice that each time the magnetic field reverses during the read interval, the bit value is read as "1", and as "0" if no reversal occurs.

The disk analog card and controller card cooperate to stack up this "serial bit stream" into an 8-bit byte, using a shift register which is the hardware equivalent of the "ASL" or "arithmetic shift left" instruction in Apple assembly language. The shift register starts out full of zeroes, and keeps schlepping in, from the right, the new bit read from the disk every 4 microseconds. The sequence in figure 10 represents the shift register contents at each of the read points shown in figure 9.

Notice that the most significant bit ("msb", or bit 7) of the shift register stays at "0" until the ninth shift, when a "1" is shifted in. This is the signal we use to decide when we should stop reading and shifting, and call it a byte. The shift register is decoded as address \$C0EC (for slot six), and the familiar instruction sequence:

```
$B954 LDA $C08C,X X=60 for slot 6
```

is used as a "wait and watch" loop to detect when the msb has finally become a one. If you are still following the discussion, you should now be able to see the reason for the first law of disk bytes (listed in Basics Part 6): If the first bit of the byte weren't a one, bit 7 of the shift register would still have a zero when we should be at the end, and we would shift at least one more time, looking in vain for a "1". The second (not more that one pair of adjacent zeroes) is required to keep the circuitry from getting lost (the third law, which requires at least one pair of adjacent ones not involving bit 7, is only for DOS 3.3, and does not affect the hardware). Let's look, for review, at some legal and illegal nibbles:

byte	binary	legal	violation
7F	0111 1111	No	Rule 1
8F	1000 1111	No	Rule 2
92	1001 0010	No	Rule 2
95	1001 0101	Yes	Not DOS 3.3
96	1001 0110	Yes	None
97	1001 0111	Yes	None
98	1001 1000	No	Rule 2
9A	1001 1010	Yes	Not DOS 3.3
9B	1001 1011	Yes	None
D5	1101 0101	Yes	*
AA	1010 1010	Yes	*

*These two bytes are not allowed in the DOS 3.3 nibblizing scheme, but are used in prologs and epilogs.

Also, tuck this away in the back of your mind: *NO* legal diskbytes can contain an 8, 1, or 0.

Now, if you want to create a non-standard disk format to keep those nasty pirates out of your "unkrackable" software, all you have to do is pick a selection of legal bytes (and maybe a few of the illegal ones), and arrange your own encoding scheme. The most common technique is an adaptation of the old encoding scheme called 4+4 nibblizing introduced to disk protection (I believe) by my good friends at Sirius software. This is the same system Apple uses to store volume, track and sector data in the address field (see Basics Part 4). As

Figure 10

Shift Register Bit #	Next Bit to be added
→ 7 6 5 4 3 2 1 0 ↓	
# of Shifts ↓	
0	0 0 0 0 0 0 0 0 <- 0
1	0 0 0 0 0 0 0 0 <- 1
2	0 0 0 0 0 0 0 1 <- 1
3	0 0 0 0 0 0 1 1 <- 0
4	0 0 0 0 0 1 1 0 <- 0
5	0 0 0 0 1 1 0 0 <- 1
6	0 0 0 1 1 0 0 1 <- 0
7	0 0 1 1 0 0 1 0 <- 1
8	0 1 1 0 0 1 0 1 <- 0
9	1 1 0 0 1 0 1 0

Figure 9

Magnetic flux direction - vs- Bit value										
Read point	0	1	2	3	4	5	6	7	8	9
Magnetic orientation	↑	↑	↓	↑	↑	↑	↓	↓	↑	↑
Resulting Bit value	0	1	1	0	0	1	0	1	0	

we described, each real byte is split into odd and even halves, and encoded so that each byte stored on the disk represents exactly 4 bits, or one nibble, of the original byte (the beginning of the disk-nibble-byte confusion). The choice for these is limited: All disk bytes are made up of A, B, E AND F, so you can have, on the disk:

AA AB AE AF BB BA BE BF
EA EB EE EF FA FB FE FF

You can find (perhaps too much) more information on this technique and decoding it in the files on Cyclod, Way Out, and Type Attack.

The files on Cyclod, Way Out and Type Attack will be published in later issues.....RDEXed

In general, while the approach to unprotecting all of these oddball formats is straight forward, the work can be long and hard, and can provide some real challenge to our skill as Krackists and programmers. In broad outline:

1. Try to figure out the disk access logic and isolate the reader/loader code.
2. Modify it to read in all the pertinent parts of the disk.
3. Save the pieces out to disk under normal RWTS structure.
4. Reconstruct the program, using as little new code as possible.

By now you should know what a loader routine looks like, and in most cases these disks will load a complete track at a time into a predetermined area of memory. By locating and altering the table of "where to read in", you can, in a few passes, read the track into memory, boot a disk, and write the memory contents out under the normal DOS format. After all the information is saved, you can begin the process of reconstruction. Usually, this consists of loading DOS (or at least RWTS) into memory and using it to manipulate sections of the original code. In cases like CYCLOD, the additional disk access between levels is totally unnecessary and can be eliminated. In something like BANDITS, however, a great deal of real data is read in at each level, and a means must be found to accommodate the disk access. Usually, this means trying to squeeze the absolutely crucial subroutines from RWTS into the space originally occupied by the loader routine.

There are a number of these "short DOS" routines in existence. These programs are all less than \$400 bytes long, and include track seek, address and data field readers, and postnibblizing routines. In the process of cracking BANDITS, both the shy "nameless" Kracker ("they said it couldn't be done...") and I wrote virtually identical routines which lived in text memory at 400-7FF. Long-John Silver has his own version of a short DOS, and an excellent implementation has recently been introduced by the Stack and The Inspector of "CORRUPT COMPUTING". It is extremely well documented, and just as in the case of Advanced Demuffin, the best utility available to the practicing Krackist.

That's a crude outline of the disk protection schemes which go beyond modified DOS; unfortunately, most are quite different in detail, and your skill as a Krackist must be made equal to the task of each one. We'll continue the basics of Kracking series next time with the long-promised article on boot-code-tracing.

The Basics of Kracking (part 8)

Boot Code Tracing

At last! The long-awaited description of boot-code tracing and its application to disk unprotection. My Kracking law #7 says "when all else fails, boot trace." For many Krackists, notable among whom was Mr. Xerox (may he rest in peace), the motto was opposite: "before you do anything else, trace the boot code." Depending on your skill and predisposition, you'll settle somewhere in between these extremes.

If Mr. Xerox didn't invent boot-tracing he was certainly the first to document it clearly in the underground press. The description that follows borrows heavily from his original treatise on the Pirate's Harbor cracking disk #1. In addition, "Mycroft" wrote a thorough article in Hardcore Computing, update 3.1, describing his own, slightly different approach to boot-tracing. While I find his process a little more laborious, it might be necessary for some very difficult cases.

The process is based firmly on the first law: track 0, sector 0 of every disk must <always> load into page 8 (\$800-\$8FF). The further assumption is that, if we can view every stage of the boot process, we can learn enough to produce an unprotected version of the program. It does not have mystical powers, and still requires the ability to tear apart and understand assembly language, much of which is intentionally misleading. We'll begin with background material and a review of the normal boot process (D@mmit, Maude, we always have to sit through the sermon first!), and proceed through an example of a new program.

(As with most Kracking activities, it's best to have on hand a blank initialized disk for saving pieces of the code as they become available.)

Ordinarily, when you boot a (DOS 3.3) 48K slave disk (a master is slightly different, but we'll ignore that for the time being), a three-stage process is started which ends up with the desired (HELLO) program running. First, the controller card ROM at \$C600-\$C6FF loads T0, S0 into page 8, then jumps to location \$801. This is a short program that loads all 10 sectors of RWTS from T0, S0 through T0, S9 into pages \$B6-\$BF (\$B600-\$BFFF), then jumps to location \$B700. This program, in turn, loads \$1B (27) pages into \$9D00-\$B5FF from T2, S4 through T0, SB (note-this is a "backwards load" for speed. Apple knew about it, so why didn't DOS ever use it for quickloading files??). After a little housekeeping, the program jumps to the DOS coldstart in \$9D84, which runs or EXECs the hello program. In summary:

Code	# of	Dest	Name
Location	Secl.	Page	Jump Next
C600-C6FF	1	08	Stage 0 801
0801-08FF	9	B6-BF	Stage 1 B700
B700-B7FF	27	9D-B5	Stage 2 9D84

Of course, in a nonstandard format intended for protection, things aren't necessarily the same. To see the differences, you need to examine each stage separately to see what it does and where it goes.

The theory of boot-tracing is straightforward: follow the boot process one step at a time to see where it leads you,

by creatively altering the code to prevent it from running away from you. In summary, we will:

1. Read in the stage 1 boot code, but not allow it to execute,
2. Alter the first stage boot so it will execute to load in stage two, while preventing the new stage from running,
3. If necessary, repeat the process of altering, loading, and halting until all the stages of the boot have been examined and understood.

In practice, the first two steps are relatively standard, but step three can get quite involved as the trace progresses.

The technique for interrupting the orderly flow of the boot is referred to as "setting break points." The terminology is borrowed from the dark ages when computers had real front panels with knobs and switches and lights, and you could actually "dial-in" an address where you wanted the computer to halt for examination (is anyone out there old enough to share my fond recollection of "execute-stop" and "fetch-stop" knobs?). Sophisticated systems with high-level executive programs still allow this today, but in the Apple we have to be a little more imaginative.

In all Apple II systems, the instruction sequence "4C 59 FF" or JMP \$FF59 goes to the reset code and provides a positive, permanent stopping place from anyplace in assembly language code, and halts with a well-defined machine state. Whenever we want to set a "breakpoint" in the Apple, we can replace any three bytes of code with "4C 59 FF".

To begin the process, let's look at some code from part of the controller card boot ROM:

```
C600:A2 20 LDX #20
C602:A0 00 LDY #00
C604:A2 03 LDX #03
...
C621:20 58 FF JSR $FF58
C624:BA TSX
C625:BD 00 01 LDA $0100,X
C628:0A ASL
C629:0A ASL
C62A:0A ASL
C62B:0A ASL
C62C:85 2B STA $2B
C62E:AA TAX
C62F:BD 8E C0 LDA $C08E,X
...
C658:A9 08 LDA #08
C65A:85 27 STA $27
C65C:18 CLC
C65D:08 PHP
C65E:BD 8C C0 LDA $C08C,X
C661:10 FB BPL $C65E
C663:49 D5 EOR #D5
C665:D0 F7 BNE $C65E
C667:BD 8C C0 LDA $C08C,X
C66A:10 FB BPL $C667
C66C:C9 AA CMP #AA
C66E:D0 F3 BNE $C663
C670:EA NOP
C671:BD 8C C0 LDA $C08C,X
C674:10 FB BPL $C671
C676:C9 96 CMP #96
C678:F0 09 BEQ $C683
...
C6E6:91 26 STA ($26),Y
C6E8:C8 INY
C6E9:D0 EE BNE $C6D9
C6EB:E6 27 INC $27
C6ED:E6 3D INC $3D
C6EF:A5 3D LDA $3D
C6F1:CD 00 08 CMP #0800
C6F4:A6 2B LDX $2B
C6F6:90 DB BCC $C6D3
```

```
C6F8:4C 01 08 JMP $0801
C6FB:00 BRK
C6FC:00 BRK
C6FD:00 BRK
```

Notice the instruction "JMP \$0801" at \$C6F8. This is the "link" to stage 1 of the boot. If we could change it to "JMP \$FF59", *every* disk we booted would load in the first sector, beep into the monitor, and obligingly wait while we snoop through page 8 to our heart's content. Since the program is in ROM, we can't alter it, but we can copy it down to a compatible location and alter it so that the program halts instead of continuing with the boot process. Because the boot code has to execute from any slot, it contains a "where are we" routine at \$C621-\$C62E to find out what its current location is. Happily for us, this kind of relocatable code will run many places besides the \$C100-\$C7FF peripheral ROM space (see the reference manual p. 81 for a description of the "where are we" routine). Mr. Xerox's famous monitor instructions which relocate the boot ROM code and insert the first breakpoint are:

```
9600<C600.C6FFM
96F8:4C 59 FF
```

(Note: page 96 is not required, but the page you use must end in 6 so that slot 6 is decoded as the controller card location). The last few lines of the (relocated) boot ROM code now read:

```
96F4:A6 2B LDX $2B
96F6:90 DB BCC $96D3
96F8:4C 59 FF JMP $FF59
```

So that typing:
9600G

will initiate a boot sequence from our code at 9600 which ends at the "breakpoint" at \$96F8, rather than continuing the boot. If you try this, you'll find that the disk is still spinning, and you can turn it off by including the instruction "2C E8 C0" (BIT C0E8) at 96F8 before the JMP \$FF59, or you can just type "\$C0E8" from the monitor. After page 8 has been loaded with the stage 1 boot code, the fun begins (until you get good at this, it's a good idea to save each piece of boot code as a BFILE on a spare disk before proceeding. It's usually easier than running through the entire sequence each time a step doesn't work as you expect, and it will make it easier to print out a disassembly of the code to figure out what it does).

At this point, page 8 must contain stage 1 of the boot with location \$801 as the starting point. If the first stage is kosher, location \$84A contains "6C FD 08", which is an indirect jump through the location in \$8FD & \$8FE. This is the exit point of the stage one boot, and normally jumps to \$B700 to begin reading in the code for stage 2 (the \$B6 at \$8FE becomes \$B7 during the 10-sector load). To continue our mission, we must locate the exit point of this stage and insert a breakpoint.

```
0801:A5 27 LDA $27
0803:C9 09 CMP #09
0805:D0 18 BNE $081F
0807:A5 2B LDA $2B
0809:4A LSR
080A:4A LSR
080B:4A LSR
080C:4A LSR
080D:09 C0 ORA #C0
080F:85 3F STA $3F
0811:A9 5C LDA #5C
0813:85 3E STA $3E
0815:18 CLC
```

```

0816:AD FE 08 LDA $08FE
0819:6D FF 08 ADC $08FF
081C:8D FE 08 STA $08FE
081F:AE FF 08 LDX $08FF
0822:30 15 BMI $0839
0824:BD 4D 08 LDA $084D,X
0827:85 3D STA $3D
0829:CE FF 08 DGC $08FF
082C:AD FE 08 LDA $08FE
082F:85 27 STA $27
0831:CE FE 08 DEC $08FE
0834:A6 2B LDX $2B
0836:6C 3E 00 JMP ($003E)
0839:EE FE 08 INC $08FE
083C:EE FE 08 INC $08FE
083F:20 89 FE JSR $FE89
0842:20 93 FE JSR $FE93
0845:20 2F FB JSR $FB2F
0848:A6 2B LDX $2B
084A:6C FD 08 JMP ($08FD)
084D:00 BRK
084E:0D 0B 09 ORA $090B
0851:07 ???
0852:05 03 ORA $03
0854:01 0E ORA ($0E,X)
0856:0C ???
0857:0A ASL
0858:08 PHP
0859:06 04 ASL $04
085B:02 ???
085C:0F ???
085D:00 BRK
...
08FE:B6 09 LDX $09,Y

```

Nonstandard formats can have any number of exit instructions, and this is where your knowledge of assembly language and experience at reading code will start to pay off. Unless the first stage is relatively standard, it's necessary to spend time examining and tearing apart the code until you understand what's going on. Look first for a jump or indirect jump to someplace outside of page 8, and change that to JMP \$FF59. If none appears, look for a "jump through the stack" trick as described in the ARCADE MACHINE file: For example, to go to \$BB00 there will be, somewhere in the code, two "PHA'S" and an "RTS". The first push onto the stack would be \$BA; the second \$FF. When the RTS is executed, the two bytes are pulled off the stack, incremented by one to \$BB00, and jumped to. In addition, more than one page can be loaded under stage 0, and accessed by a relative branch instruction, so you'll have to examine <all> the code loaded in (it's good practice to clear out all of memory before starting; this will work if DOS is not active: 800:0 N 801<800.BFFFM).

When you find the exit point, make it a breakpoint with "4C 59 FF" to prevent the continuation of the boot. Before proceeding, take a good look at all the code to be sure you understand where the next stage loads, and any unusual conditions or instructions.

The altered portion of code is now:

```

0839:EE FE 08 INC $08FE
083C:EE FE 08 INC $08FE
083F:20 89 FG JSR $FE89
0842:20 93 FE JSR $FE93
0845:20 2F FB JSR $FB2F
0848:A6 2B LDX $2B
084A:4C 59 FF JMP $FF59
084D:00 BRK

```

The theory now is to allow the boot to proceed through one more stage, halting after RWTS has been read in, and giving us a chance to examine that portion of the program for alterations. If we just rebooted with "9600G", the original code would overwrite our altered page 8, so

we have to arrange it so that the first stage boot code is sent off into oblivion. Referring back to the boot code, location \$9658 (originally \$C658) contains the page number where T0, S0 loads in, normally 08. Changing it to \$20 will cause T0, S0 to load into \$2000 instead of \$0800, and the boot will continue through our altered page 8. Note that we have to remove the first break point at \$96F8 and restore the original JMP \$0801:

```

9658:20
96F8:4C 01 08

```

Now, when we type "9600G", the boot code will load T0, S0 into \$2000-\$20FF, where it won't bother us at all, then jump to \$801 to execute our code. After RWTS has been loaded in, instead of jumping to \$B700 to continue loading DOS, the program hits the (second) break point at \$84A and halts.

The final phase of this process is to locate the exit point from this area of code, insert another breakpoint, and examine all the code loaded in by stage 2. Again, we have to make sure that the boot process doesn't overwrite the changes, which means we have to understand how the destination addresses are set up in stage 1. Even in normal DOS it's not obvious, but enough head-scratching or reading of Beneath Apple DOS will reveal that the byte in location \$8FE is one higher than the first page loaded into, and the byte at \$8FF is one less than the number of sectors to be loaded. As before, we remove the previous breakpoint, alter the destination of the real code loaded in under this stage, and set the new breakpoint:

```

B700:8E E9 B7 STX $B7E9
B703:8E F7 B7 STX $B7F7
B706:A9 01 LDA #$01
B708:8D F8 B7 STA $B7F8
B70B:8D EA B7 STA $B7EA
B70E:AD E0 B7 LDA $B7E0
B711:8D E1 B7 STA $B7E1
B714:A9 02 LDA #$02
B716:8D EC B7 STA $B7EC
B719:A9 04 LDA #$04
B71B:8D ED B7 STA $B7ED
B71E:AC E7 B7 LDY $B7E7
...
B738:20 93 B7 JSR $B793
B73B:A2 FF LDX #$FF
B73D:9A TXS
B73E:8E EB B7 STX $B7EB
B741:4C C8 BF JMP $BFC8
B744:20 89 FE JSR $FE89
B747:4C 84 9D JMP $9D84

```

The changes are:

```
84A:4C 00 B7
```

(We can't use the indirect jump in the original, since we have redirected the boot.)

```
8FE:20 09
```

(Page 20 or anyplace else where 10 pages of code won't hurt anything.)

```
B747:4C 59 FF
```

JMP \$9D84 is the DOS cold-start. The JMP \$BFC8 is a patch which returns with a JMP \$B744)

The last few lines of code are now:

```

B741:4C C8 BF JMP $BFC8
B744:20 89 FE JSR $FE89
B747:4C 59 FF JMP $FF59

```

Now type 9600G, and let's recap the process that will occur:

The modified stage 0 code at 9600-96FF will load T0, S0 into page 20 (since we don't want it), then jump to the start of our modified page 8 at 801.

The modified page 8 will load T0, S0 through T0, S9 into pages 20 to 29, then jump to our modified code at \$B700.

The modified code at \$B700 will load 27 sectors of DOS into pages \$9A-\$B5, then halt when it hits the breakpoint at \$B747.

Before the next episode, try this process on a few different disks, including some protected ones - practice is essential. In the second part, we'll take on SSI'S RDF 1985.

John C. De La Cruz CA

Softkey for...

Rendezvous with Rama Fahrenheit 451

Trillium

Requirements:

2 blank disks
A fast copy program
Copy II plus sector editor

1. Copy original disks to the blank disks with fast copy program such as Locksmith 6.0.
2. With your sector editor scan side A of the copy for 20 30 E0.

Rendezvous with Rama

Trk	Sct	Byte	From	To
\$03	\$0C	\$24-26	20 30 E0	20 29 1C

Fahrenheit 451

Trk	Sct	Byte	From	To
\$02	\$0C	\$28-2A	20 30 E0	20 2D 1C

3. Don't forget to write the sector back to the disk.

Softkey for...

Magic Spells v2.0

The Learning Company

Requirements:

A blank disk
A sector editor
A fast copy program

1. Make a copy of your original
2. Scan disk for the following bytes 08 68 8D 5E 61 78 A9 Change the \$08 to \$60. Here is where I found the bytes on my disk.

Trk	Sct	Byte	From	To
\$14	\$09	\$00	08	60

3. Write the sector back to the disk.

Softkey for...

Magic Spells v2.1

The Learning Company

Requirements:

A blank disk
A sector editor
A fast copy program

1. Make a copy of the original.
2. Scan disk for the following bytes A2 03 B5 00 48 CA 10 FA Change the \$A2 to \$60. Here is where I found the bytes on my disk.

Trk	Sct	Byte	From	To
\$14	\$00	\$00	A2	60

3. Write the sector back to the disk.

Softkey for...

Teacher's Tool Kit

Word Search 3.1
Word Match 3.1
Word Scramble 3.1
Multiple Choice 3.1

Hi Tech of Santa Cruz

Requirements:

Apple IIgs
Copy II plus on 3.5" disk

4 blank 5.25" formatted ProDOS disks with volume name /HI.TECH.BOOT

1 blank 3.5" formatted ProDOS disk or RAMdisk on IIgs

1 5.25" disk with normal ProDOS, BASIC.SYSTEM on it.

The Teacher's Tool Kit comes on four 5.25" disks that have a non-standard ProDOS format with altered trailers. I tried to make a bit copy of it, and even used COPYA to make a copy with standard trailers, but neither copy would boot. On Computist #65 page 18, Jack Moravetz's softkey helped. But his softkey was for an older release of the Teacher's Tool Kit. So the resulting copies still would not work. On my copy of the tool kit each of the four programs come on a two sided disk, the front side is the program disk and is unprotected, the back side is the startup disk which is protected with the non-standard trailers. So to boot up the disk you flip it over boot the computer then you will be prompted to insert the program disk or press ESC to quit.

Boot up one of the programs and when prompted to insert the program disk. Press ESC to quit. Their ProDOS has a Quit code that will show you all the volumes in your 3.5" drives, RAMdisk, and even their protected disk. But if you have a normal ProDOS disk in your 5.25" drive it won't read it. Insert Copy II+ plus in your 3.5" drive and Prefix that drive, now you will be shown all the files on Copy II+, from here select and run UTIL.SYSTEM. Remove Copy II+ and insert your formatted 3.5" disk and use the copy files option to copy the following files from the startup side of all four disks to your disk:

Word Search: WSBT.SYSTEM and WS.BTPROG
Word Match: HTP.SYSTEM and PARAMS.BT
Word Scramble: SCBT.SYSTEM and SCPARAMS.BT
Multiple Choice: MC.SYSTEM and PARAMS.BT

If you noticed on Word Match and Multiple Choice that the second file is named the same, then on Multiple Choice, rename the second file to PARAMS.BT1.

Next let's get rid of their ProDOS now that we're done with it and boot up Copy II+ and copy the files to the four 5.25" formatted disks with the volume /HI.TECH.BOOT and label each disk accordingly, remember to rename PARAMS.BT1 back to PARAMS.BT on Multiple Choice. Copy the Program Disk side to the other side of each disk.

Now to remove the copy protection boot BASIC and insert each disk and load the following files:

Word Search

BLOAD WSBT.SYSTEM, TSYS, A\$2000
CALL-151

2063:60 18 60 was 08 38 2C
BSAVE WSBT.SYSTEM, TSYS, A\$2000

Word Match

BLOAD HTP.SYSTEM, TSYS, A\$2000
CALL 151

2063:60 18 60 was 08 38 2C
BSAVE HTP.SYSTEM, TSYS, A\$2000

Word Scramble

BLOAD SCBT.SYSTEM, TSYS, A\$2000
CALL 151

2063:60 18 60 was 08 38 2C
BSAVE SCBT.SYSTEM, TSYS, A\$2000

Multiple Choice

BLOAD MC.SYSTEM, TSYS, A\$2000

CALL-151

2054:60 18 60 was 08 38 2C

BSAVE MC.SYSTEM, TSYS, A\$2000

That's all enjoy your unprotected copies.

Softkey for...

Teacher's Tool Kit (IIC)

Word Search 3.1

Word Match 3.1

Word Scramble 3.1

Multiple Choice 3.1

Hi Tech of Santa Cruz

Requirements:

Apple IIC

4 blank 5.25" formatted ProDOS disks

with volume name/Hi.TECH.BOOT

1 5.25" disk with normal ProDOS, BASIC.SYSTEM on it.

The easiest way to unprotect these disks is if you have a IIGs with a 3.5" drive or a RAMdisk. But I realize not everyone has a IIGs so I decided to figure out a way to retrieve the necessary files from the protected side on my Apple IIC. On the IIC it's not so easy. It involves boot code tracing, since I couldn't reset out of the program without trashing the program in memory.

For the hard way on the IIC, enter the Monitor and clear memory by filling it with zeros.

CALL-151

800:00 N 801<800.BFFFFM

Next move the boot code down to where we can alter it.

8600<C600.C7FFM

Now enter the following

863A:20 09 87

8728:A9 88

86F8:4C 01 88

What we've done here is change the boot code (Boot 0) so that the next Boot stage (Boot 1) will load at page \$8800 instead of \$0800 insert one of the original disks in drive 1. After making the changes check to see that they are correct, then run the code at \$8600 (Boot 0).

8600G

Stop the drive by typing:

COE8

Check memory pages 8800 and 8900 there should be some code at both locations especially in page \$8900 to \$89FF. If page 8900 is still filled with zeros then redo the above steps over again. If the code is there then we have to move both pages at \$8800, \$8900 to \$0800, \$0900.

800<8800.89FFM

After moving the code we have to do a couple of changes. First if you look at the Boot 1 code, at location \$08FC there is a JMP \$2000 which is the jump to start ProDOS which we want to stop from executing. Here I put JMP \$FF59 which puts us back into Monitor. And the second change is in Boot 0 at location \$86F8 where we just change it back to its original value of JMP \$0801.

8FC:4C 59 FF

86F8:4C 01 08

Restart Boot 0 again.

8600G

When the drive stops, ProDOS will have been loaded into memory at \$2000 now we want ProDOS to load the SYSTEM file (example HTP.SYSTEM) but not to start it. So we make another change this time on the ProDOS in memory. At location \$2316 there is a JMP \$2000 which is the jump to the start of the SYSTEM file. Here we enter another

JMP \$FF59 to prevent the SYSTEM file from executing and to throw us back into the monitor.

2316:4C 59 FF

Now run the ProDOS in memory.

2000G

When the drive stops you'll be back in the Monitor again. With the SYSTEM file in memory. Up to this point the above information will work on all four disks past this point I'll give you the different information for each disk.

Next we have to move the SYSTEM file to a safe memory location.

For Word Search, Word Match, and Word Scramble enter:

8000<2000.25FFM

For Multiple Choice enter:

8000<2000.24FFM

Next there is one more file that has to be loaded in from the disk but again we want it loaded and not executed.

For Word Search and Word Match enter:

20F4:4C 59 FF

For Word Scramble enter:

20F7:4C 59 FF

For Multiple Choice enter:

20C4:4C 59 FF

What we've done is change the JSR 2000 BF 65 which is the JMP to the Quit Code in ProDOS. And changed it to drop us back into the Monitor. Next run the SYSTEM file.

2000G

When the Title screen comes on to insert the PROGRAM DISK or press ESC to Quit. Press ESC do not press any other key. Pressing ESC would normally take you to the Quit Code but with the change we made above you will be put into the Monitor. Once in the Monitor do the following to move the file that was loaded into memory to a safe place where it won't be erased.

For Word Search enter:

8700<B400.B58AM

For Word Match and Word Scramble enter:

8700<B400.B4FFM

For Multiple Choice enter:

8500<AC00.AFFFM

Now that both files are in a safe location. Remove the original and insert the disk with the normal ProDOS and BASIC.SYSTEM, to get you into BASIC. Once your in BASIC type CALL-151 to enter the Monitor. Now let's move the files back to where they belong so we can save them to disk.

For Word Search enter:

2000<8000.85FFM

B400<8700.888AM

For Word Match and Word Scramble enter:

2000<8000.85FFM

B400<8700.87FFM

For Multiple Choice enter:

2000<8000.84FFM

Since you are in the Monitor already, you can unprotect the SYSTEM file.

For Word Search, Word Match, and Word Scramble enter:

2063:60 18 60

For Multiple Choice enter:

2054:60 18 60

Get back to BASIC (enter 3DOG) and BSAVE the files.

Word Search

CREATE WSBT.SYSTEM, TSYS

BSAVE WSBT.SYSTEM, TSYS, A\$2000, L\$600

BSAVE WS.BTPROG, A\$B400, L\$18A

Word Match

CREATE HTP.SYSTEM, TSYS

BSAVE HTP.SYSTEM, TSYS, A\$2000, L\$600

BSAVE PARAMS.BT, A\$B400, L\$100

Word Scramble

CREATE SCBT.SYSTEM, TSYS

BSAVE SCBT.SYSTEM, TSYS, A\$2000, L\$600

BSAVE SCPARAMS.BT, A\$B400, L\$100

Multiple Choice

CREATE MC.SYSTEM, TSYS

BSAVE MC.SYSTEM, TSYS, A\$2000, L\$500

BSAVE PARAMS.BT, A\$8500, L\$400

And that all there is to it so enjoy.

By the way, if you like their Quit Code (which I did) you can capture it to use it with your own ProDOS. When you have their ProDOS in memory move the Quit Code to a safe place in memory.

8000<5700.59FFM

Boot a normal ProDOS and get into BASIC. Then:

BSAVE QUIT CODE, A\$8000, L\$300

Now you can load in your normal ProDOS and patch it.

BLOAD PRODOS, TSYS, A\$2000

BLOAD QUIT.CODE, A\$5A00

BSAVE PRODOS, TSYS, A\$2000

Note: Different versions of ProDOS have the Quit.code at different locations. On versions 1.2, 1.3, 1.4, 1.6 and 1.7, it lives at \$5900 so on these versions BLOAD the QUIT.CODE at A\$5900. On version 1.1.1 it lives at \$5700 and on version 1.8 it lives at \$5A00.

Softkey for...

Fay's Word Rally

Didatech

Requirements:

A blank disk

A sector editor

Locksmith 6.0 or fast copy program that will ignore errors

1. Make a backup using Locksmith fast-copy to ignore error on track 2, sector 7.

2. Use your sector editor to make the following sector edits:

Trk	Scr	Byte	From	To
\$00	\$04	\$89-91	A2 1C 98 9D	A9 D0 8D F6
			E0 BC CAD0	86 8D 96 7A
			FA	EA

3. Write the sector back to the disk.

Softkey for...

Monsters and Make Believe v1.1

Dinosaur Days v1.0

Pelican Software

Requirements:

Apple IIGs with ROM v01 & up

FID from the DOS 3.3 System Master

2/3 blank disks formatted with DOS 3.3

Monsters and Make Believe uses DOS 3.3 but the disk is formatted in a very strange format. The DOS on this disk can read itself and also a normal DOS 3.3 disk. This was easy to figure out since the data disks that it creates are in a normal format. I could not copy this disk with any of the disk copiers or nibble copiers that I have. And everything else that I tried didn't work either. I gave up on this program at one point. But later I gave it another try and finally succeeded. What I did was use the IIGs's

built in CDA Visit Monitor option to snoop around after the program was loaded in memory and found that most of Monsters and Make Believe is written in BASIC. After trial and error I finally came up with a way to copy the files to a normal disk using their protected DOS with FID.

1. Boot up your IIGs and enter the Monitor with a CALL-151, at the asterisk type the # sign and return to the get the CDA Visit Monitor installed.

2. Next on one of the blank disk copy FID onto it then type this short BASIC program and save it to the same disk with the name MAKE. If you only have one drive change D2 to D1.

```
10 TEXT : HOME
```

```
20 PRINT CHR$(4) ; "BRUN  
FID, D2"
```

3. Boot up the original Monsters and Make Believe, when you get to the main menu Press Control, Open Apple, ESC to get to the Control Panel from here choose Visit Monitor. Once in the Monitor type the following bytes.

101B:4D 41 4B 45 2C 44 32 22 3A

What you just did was change part of the BASIC program in memory from "RUN SETUP, D1" to "RUN MAKE, D2": again if you only have one drive change the 32 to 31.

Then press ctrl Y and press return to reenter the Control Panel. Choose Quit to return to the program. Remove the original disk if you only have one drive and insert the disk with FID and the short BASIC file on it. If you have two drives leave the original in drive one and insert the disk with FID into drive two. After you have done this choose Change Setup from the main menu which will run the MAKE file we created to brun FID.

4. Once FID is loaded, put the original back into the drive choose COPY FILES and copy all the files from the original to your formatted disk.

That's all there is to it. Easy isn't it.

Softkey for...

Same or Different

Learning Technologies

Requirements:

A blank disk

A sector editor

A fast copy program

I used the Softkey from Computist #66, page 8 to help me crack this one.

1. Fast copy your original to your blank disk.

2. Make the following sector edits to your copy:

Trk	Scr	Byte	From	To
\$06	\$04	\$8A-8B	D0 D8	60 EA

3. Write the sector back to you disk.

Softkey for...

Greeting Card Maker

Activision

Requirements:

A blank disk

A fast copy program

A sector editor

1. Copy the original to your blank disk.

2. Make the following sector edits:

Trk	Scr	Byte	From	To
\$20	\$09	\$58-5C	??	A9 FF 18 90 3D

3. Write the sector back to the disk

I'm sorry but when taking down my notes I neglected to write down the original bytes that I found.

Softkey for...

GATO v1.3

Spectrum Holobyte

Requirements:

- COPYA
- Blank disk
- Protected GATO v1.3 disk
- A sector editor

I followed the softkeys from Computist #59 and #62 by Mike Egnotovich and I was able to transfer the files to a normal Pascal formatted disk but the copy still wouldn't boot. It would grind and then stop with the message "INSERT BOOT DISK WITH SYSTEM.PASCAL ON IT, THEN PRESS RETURN". I found that if at this point I inserted the original GATO 1.3 disk that the program would load and start as if I had booted up the original. I finally figured out that maybe the boot file was looking for the protected header and trailers on the original GATO. So I scan the disk for C9 6A and I hit the jackpot.

1. Boot your DOS 3.3 system disk.
2. Change DOS and make a COPYA copy of your Gato 1.3 disk.

CALL-151

B954:4A C9 6A D0 EF

B98B:18 60

RUN COPYA

3. Use the System Utilities to format a disk in Pascal and copy the files from the COPYA copy to the disk you formatted.

4. Make the following sector edits:

Trk	Sct	Byte	From	To
\$00	\$0F	\$71	AF	DE
		\$7A	08	AA
		\$34	4A	EA
		\$36	6A	D5

5. Write the sector back to the disk.

Zorro FL

Questions & Answers

To E.N. Hondrick: Thanks for your information published in COMPUTIST #77. However, concerning FrEdwriter, please be aware that I was not attempting to sell copies of the program, nor did I wish to make a dishonest profit - I'm well aware that the program is 'Free-ware'; I merely used the term 'public domain' in a very general fashion. Believe me, I know how some programs which end up in public domain can appear under several different authors, each asking for a fee as if the program were Shareware.

As to your offer of that EPROM burner, thanks, but lately I've been trying to raise some cash to get an IBM clone (GASP!), a situation which is somewhat complicated by the fact that I don't wish to sell my Apple IIe, not to mention my current financial situation. However, even if I did have the money, I'm sorry, but I'd still have to say no, because there have been some ads I've looked through recently offering burners (with docs) for less than \$50. Thanks anyway, though.

To Bob Igo: Thanks for your inquiry in COMPUTIST #76. It's been awhile, but if I remember correctly, I got a DANGEROUS rating on Elite the quick and hard way; I traded illicit goods in Anarchic and Feudal worlds, and destroyed any pirates that got in my way. After doing this for a long, LONG time, I eventually got DANGEROUS status. It seems that the number of ships you need to reach the next combat rating is

exponentially greater than the previous one; for example, 2 ships from Harmless to Mostly Harmless, 4 ships (2 squared) from Mostly Harmless to Poor, etc. Of course, these numbers are just wishful thinking, because in actuality it seems that you need many more kills to advance than just 2 or 4 per rating.

The Elite disk, as you suggested, is heavily encoded. After first receiving that message about the lost ship, I tried to re-read it by using the Copy II Plus sector editor to scan for the text, but to no avail; it, and anything else in English on the disk were obviously encoded. Luckily, I recently was able to attain a COMPETENT rating again (with the help of Dan Reid's APT in COMPUTIST #75, and with the assistance of Commander Jameson), and thus re-read the bulletin. I even took a photo of the entire message, so for your benefit, and the benefit of any other Eliters, I'm writing it here:

"Greetings, Commander JAMESON, I am Captain Curruthers of Her Majesty's Space Navy and I beg a moment of your valuable time.

We would like you to do a little job for us.

The ship you see here is a new model, the Constrictor, equipped with a top secret new shield generator.

Unfortunately, it's been stolen.

It went missing from our ship yard on Xeer Five months ago and was last seen at Reesdice.

Your mission, should you decide to accept it, is to seek and destroy this ship.

You are cautioned that only Military lasers will penetrate the new shields and that the Constrictor is fitted with an E.C.M. system. Good luck, Commander."

I'd be glad to help in discovering the secret to attaining Elite status, so please send in any ideas you've got on how to organize some sort of APT collection for Elite.

To Jeff Hurlburt: Thanks for your tips on Elite within COMPUTIST #76. After reading through them, though, I still have a couple of questions to ask of you. First off, I know that to complete the first mission for the Navy, I have to seek and destroy the Constrictor. Could you tell me in which galaxies the planets Xeer Five and Reesdice are, and if you can, could you tell me their general location within the galaxy on the galactic starmap? Also, could you give the general location of Errius in galaxy 2? I'd really appreciate your help on this. And by the way, your poem in COMPUTIST #76 was great - maybe a bite on that citric deluge will tell lame software producers to get going on some better stuff!

To Groucho: I enjoyed reading your tips for Wasteland in COMPUTIST #75. There's just one part of the game I'm stuck on, so I'll take up your offer and ask for some advice.

After wasting Finster at Darwin Base, I salvaged his head and had my character with Cyborg skill install it in the mindlink. I then had this same character don the helmet and enter what I assume to be Finster's perverted brain. I easily made it past the first, second, and third rooms, but I find myself hopelessly stuck in the fourth, since the unruly floor keeps

my character from reaching the console. I recall that I once made it through, after Finster said something like "well, that was quite a logic flaw", and then the floor ceased its annoying conveyor-belt action. However, even though I saved the game, I found I couldn't make it through again, because after hacking the Proton Axe vainly against the web of deceit in room five, I switched characters, and re-entered the labyrinth with a different character, who is now unable to reach the console in the fourth room. I'd appreciate any tips on how to get through the fourth and fifth rooms.

To Gary Wills: Concerning your 3.5" modification to OMEGA, I might be able to help you out if I knew what procedure you used to attain the "limited success" that you've had. I've studied the formats of Ultima V and Windwalker, as well as OMEGA, and each seem to use the same DOS, namely DinkeyDOS. I have for quite a while been trying to make these programs upgradeable, but since the only manual on ProDOS I have is too general, I haven't been able to thoroughly develop a procedure (Beneath Apple ProDOS is next on my "to buy" list, for sure). What I was thinking of doing is to somehow rename the files and volumes, and then alter them with a sector editor so that the program treats each separate disk side as a subdirectory, instead of a volume. As you can see, the process I've got in mind is not quite refined! Also, have you noticed that the file DINKEYDOS on boot side of OMEGA and the other programs seems wholly responsible for booting the program? It doesn't even utilize any other SYSTEM-type files (i.e. BASIC.SYSTEM, etc.).

Ⓢ I hope that someone out there can provide a detailed breakdown of the file DINKEYDOS before I strain my brain trying.

To Ross A. Holmes: In answer to your query in COMPUTIST #75, concerning that "program" Little Computer People by Activision. I first saw this program way back in 1983, and it is really just a silly little demonstration program in which the authors wished to personify the bytes within your Apple, and have you watch them "live". It's somewhat similar to watching guppies in an aquarium, but it's a lot less interesting.

To Edison: I know Ultima III like the back of my hand, so either contact me through the RDEXed, or send a letter to COMPUTIST, and I'll answer as soon as possible.

Steven Kalynuik Canada

First, I would like to thank the reader's of COMPUTIST who came to my aid in request of the programs COPYA and DISK MUNCHER.

However my request list was not completely fulfilled as I still require the following programs:
DOS TOOL KIT ver1.0 backside by Apple Computer (my HRCG, RLOAD, RBOOT don't work)
WIZARD'S WORKBENCH I or II
WIZI-SCOUT
WIZI-DOC

All the "WIZ" programs are from the now nonexistent company of Magicsoft who ripped my off as they cashed a money order for WIZARD'S WORKBENCH II and never sent it, on trying to contact them I found out that on the day

after they cashed in my money order they claimed bankruptcy.

These programs are Wizardry scenario creators and editors.

To Blain Johnson: Thank-you for your softkey for ELECTRONIC ARTS SOFTWARE in COMPUTIST #76. This softkey, without modification, allowed me to copy Wasteland. The copy boots perfectly and now the original is safe. The copy even boots quieter. Unfortunately the program could not copy LEGACY OF THE ANCIENTS.

Ⓢ Does anyone know why the latest version of COPY II PLUS (ver 9.0) does not accept the RAM disk option for choice of drives. I cannot FORMAT the RAM drive from the menu. The COPY II PLUS (ver 8.3) accepts it and I use it all the time.

Ⓢ Does anyone know how to make the view files option of COPY II PLUS in general, print 80 columns to printer instead of just 40 columns like it does.

Ⓢ Why is "Wizardry: Return of Werdna" (Wizardry IV) on the most wanted list. This program, at least my copy is only protected by a password check, no disk protection, the same is true for Wizardry: Heart of Maelstrom (Wizardry V) only password check. Could it be possible that it is on here for password removal, if so the please add Wizardry V: Heart of Maelstrom as the password check is annoying.

When I asked if password (and other non-disk) protection should be handled the same as disk protection, the readers overwhelmingly responded "YES". So any kind of protection makes software eligible for the most wanted list..RDEXed

I myself even with the password checks have finished all Wizardry's with no cheat disks or editing. I am now awaiting an Apple IIe or IIc version of Wizardry: Bain of the Cosmic Forge. As of January, Sir Tech the makers of Wizardry said that only a MSDOS version will be out.

Outrageous! First ORIGIN made the mistake of not making an Apple IIc or IIe version of Ultima VI: The False Prophet. This disappointed a lot of Apple owners who bought all or most of the Ultima series, awaiting more, just to then be left in the cold by the company they supported. Now Sir Tech looks like they are doing the same thing.

Somehow let us Apple owners make them (the companies) acknowledge the Apple computer is alive and well and existing on the planet Earth, not some distant world, as companies now seem to think. Let's see more Apple IIc, IIe programs in the future.

IBM RDEX IBM RDEX

Don Westcott CO

I recently bought a DTK 386SX IBM compatible computer so I could play some of these MS-DOS games that aren't being released for APPLE.

Ⓢ One of my favorites is SIM CITY by MAXIS. I tried the softkey for SIM CITY on page 22 of COMPUTIST #74. Using PCTOOLS I found the byte string and made the alteration but when I ran it I was still asked for information from the unreadable chart. Does anyone have another softkey for SIM CITY I could try?

⊕ Another one that I like is SIERRA's new STELLAR 7. It plays very similarly to the old APPLE game but the graphics are much better and it has no copy-protection! I also like STORMOVIK from ELECTRONIC ARTS but it has a tedious manual check protection. You have to look up the English translation of a Russian word. I hope someone has a softkey for this.

My IBM Most Wanted List

Ace of Aces	Accolade
Battle Chess II	Interplay
Colony, The	Mindscape
Heat Wave	Accolade
Sim City	Maxis
Stormovik	Electronic Arts
Tower Toppler	U.S. Gold
Wayne Gretzky Hockey 2	Bethesda

Marc Batchelor FL

IBM Softkey for...

Welltris

Spectrum Holobyte

When I read the crack for this piece of software in Computist number 74 (Mike Basford), I decided to try it out since I happened to be attempting to crack it myself. His crack DID allow you to get past the opening screen without ever asking for name of some city from the stupid manual. However, the crack (on my version) was bad in that the vertical lines on the pit (for lack of a better term) disappeared. Without these lines, it is very difficult to know where the pieces are going to fall. So, I continued with my crack.

Where is it?

Tracing the code through its opening screen was indeed fun, having to deal with graphics and all. However, using Borland Software's Turbo Debugger made it much easier. I found the call to the protection (similar to Mike's) around \$1B50 (that is Hex 1B50). Following is a listing of the surrounding code:

```
1943:1B4A 50 PUSH AX
1943:1B4B E8FEE5 CALL 0140
1943:1B4E 44 INC SP
1943:1B4F 44 INC SP
1943:1B50 E86942 CALL 5DBC Call to
Protection
1943:1B53 E8BD15 CALL 3113
1943:1B56 E99F00 JMP 1BF8
```

Mike's crack involved NOPing the call to the protection. However, in my version, the "protection" routine also sets up some things (or calls some routines) that are responsible for the drawing of the vertical lines.

Diving right into the thick of things, I listed out the code at 5DBC until I came across the following code:

```
1943:6034 E896DE CALL 3ECD
1943:6037 83C408 ADD SP,+08
1943:603A E8A6D8 CALL 38E3Copy
protection screen
1943:603D C646ED02 MOV BYTE PTR [BP-
13],02
...
1943:6073 FFB71A03 PUSH [BX+031A]
1943:6077 E8E918 CALL 7963 Clear box for
input
1943:607A 83C40A ADD SP,+0A
...
1943:60A2 50 PUSH AX
1943:60A3 E88B17 CALL 7831 Clear box for
message
1943:60A6 83C406 ADD SP,+06
...
1943:60C6 50 PUSH AX
```

```
1943:60C7 E89FA3 CALL 0469 Get answer for
question
1943:60CA 83C40C ADD SP,+0C
1943:60CD 33C0 XOR AX,AX
1943:60CF A39410 MOV [1094],AX
1943:60D2 A39210 MOV [1092],AX
1943:60D5 EB42 JMP 6119 Verify answer
```

At this point, I NOP'ed out the calls to the screen routines, and started down the merry path of examining the verification routine. After investigation (tracing), it became obvious that the code does not ever verify the complete answer, but only the first four characters (lazy programmers!). This fact however doesn't help our effort to crack this one. At 1943:6119, the code checks its index variable ([1092]) for the value 4 (as noted above), and if the index is less than that, it jumps to 60DD (thru 60D7). Following the code here revealed that the location of the user entered answer could be obtained through subtracting 2C from the byte pointer (BP) + the index number ([1092]), and that the expected answer is loaded byte by byte into the DX register. Also, the code sets a pass flag ([BP-17]) to FF if any one of the 4 characters expected do not match the answer provided by the user. That chunk of code is listed as follows:

```
1943:60EA 8D5ED4 LEA BX,[BP-2C] WRITE
THIS DOWN
1943:60ED 031E9410 ADD BX,[1094]
1943:60F1 36 SS:
1943:60F2 8A07 MOV AL,[BX]
...
1943:60F7 E86841 CALL A262Don't know
purpose?!
1943:60FA 44 INC SP
1943:60FB 44 INC SP
...
1943:6103 26 ES:
1943:6104 8A5702 MOV DL,[BX+02] The
Expected Value
1943:6107 B600 MOV DH,00
1943:6109 3BC2 CMP AX,DX
Compare Ours
with theirs
1943:610B 7404 JZ 6111 If equal, skip
bad flag
1943:610D C646E9FF MOV BYTE PTR [BP-
17],FF Set Bad
1943:6111 FF069210 INC WORD PTR [1092]
1943:6115 FF069410 INC WORD PTR [1094]
1943:6119 833E921004 CMP WORD PTR
[1092],+04
1943:611E 7CB7 JL 60D7
1943:6120 807EE900 CMP BYTE PTR [BP-
17],00 Check
Flag
1943:6124 7503 JNZ 6129 Failure
1943:6126 E90201 JMP 622B Must be ok,
continue.
1943:6129 33C0 XOR AX,AX Start
Failure routine.
```

When I saw the above code, I thought that this would be as simple as flipping the FF on line 601D to a 00, and letting 'er rip. However, trying this approach results in some message at the top of the screen (something like "Escape Code 0x0F7") and then an abnormal termination. So, I was forced to patch in some code that copies the expected answer to the location of the actual answer. The only relevant question left to answer is where to put such a routine. After much needless deliberation, it came to me... why don't we use 6129 since it won't ever be used again (we won't be failing the check again). This turns out to work perfectly. The code that does just that is listed below:

```
1943:6129 032E9210 ADD BP,[1092] BP +
Index
1943:612D 36 SS:
1943:612E 8856D4 MOV [BP-2C],DL Move to
ours, theirs
1943:6131 3E DS:
1943:6132 2B2E9210 SUB BP,[1092] BP -
Index
1943:6136 C3 RET
```

Finally, we need to replace the compare on line 6109 with a CALL to our routine, and replace the FF on line 6110 (bad flag) with 00 (good flag), and NOP the exiting calls to the graphics that clear out the message box (CALL 7963 on line 62C1 - Not shown) and display "That is correct" (CALL 3F4B on line 62EB - Not shown).

Step by Step

Note: The following steps are provided for the "non-debug literate". For those of us who are, remember the days when you asked: "What's a debug?" We have to give the beginners an opportunity to liberate their software as well.

1. Rename and copy the WELLTRIS.EXE file:
COPY WELLTRIS.EXE WELLTRIS.BAK
RENAME WELLTRIS.EXE WELLTRIS
2. Fire up DEBUG and prepare to make some modifications:
DEBUG WELLTRIS
3. Search for the CALL 38E3 (Line 603A shown above)
S0 FFFF E8 A6 D8
4. The system will respond with:
XXXX:YYYY
Issue the following debug command:
E yyyy 90 90 90
5. Search for the CALL 7963 (Line 6077 shown above)
S0 FFFF E8 E9 18
6. Issue the following debug command (using step 4 as a guide)
E yyyy 90 90 90
7. Search for the CALL 7831 (Line 60A3 shown above)
S0 FFFF E8 8B 17
8. Issue the following debug command (using step 4 as a guide)
E yyyy 90 90 90
9. Search for the CALL 0469 (Line 60C7 shown above)
S0 FFFF E8 9F A3
10. Issue the following debug command (using step 4 as a guide)
E yyyy 90 90 90
11. Search for the check routine (Line 6107 shown above)
S0 FFFF 3B C2 74 04 C6 46 39 FF
12. Issue the following debug command:
E yyyy E8 1D 00 90 C6 46 E8 00
13. Search for the fail routine (line 6129 shown above)
S0 FFFF 33 C0 50 B8 1E 00 50 A1
14. The system will respond:
XXXX:YYYY
XXXX:ZZZZ
Issue the following debug command:
E yyyy 03 2E 92 10 36 88 56 D4 3E 2B 2E 92 10 C3
15. Search for the CALL 7963
S0 FFFF E8 9F 16
16. Issue the following debug command:
E yyyy 90 90 90
17. Search for the CALL 3F4B
S0 FFFF E8 5D DC
18. Issue the following debug command:
E yyyy 90 90 90
19. Write out all the changes with the following debug command:

W

20. Exit debug.

Q

21. Rename the file WELLTRIS to WELLTRIS.EXE
REN WELLTRIS WELLTRIS.EXE

That concludes the crack for Welltris (dated 10-3-89). I hope the above article has been helpful to those who are aspiring IBM crackers.

IBM Softkey for...

Serve and Volley

Accolade

Serve and Volley (S&V) by Accolade is a very fun game tennis simulation. The protection used involved self-modifying code that turns normal Interrupt 21's (Dos service calls) into ferocious Interrupt 13's (disk drive service calls). When started up from a hard drive (or a copy for that matter), a routine in the main program performs a search of all drives for the 'key' disk. If this disk is not found, it jumps directly into demo mode.

Tracing the code using Turbo Debugger (Borland Software) was very easy since there were no "funky" far returns (RETF) and very little messing with the stack. I found the entry point to the program around \$B53 (hex 0B53). Following this chunk of code led to a CALL 9053 (E8 DD 84). After tracing this code (very messy!), I found that it modified the INT 21's to INT 13's, read several sectors off the drive (for no apparent reason), and returned with the AX register zero'd out. Upon return from the routine, the AX register is tested for zero, and if it is zero, hex 0000 is stored at a memory location ([203A]).

The solution is a 3 byte fix. Instead of CALLing 9053, we store 0000 into the AX register. This passes the protection routine without ever looking for a key disk.

Step by Step

1. Copy all files from the original disk to you hard drive (or another disk)
C:
CD TENNIS
XCOPY A:.*
2. Rename TENNIS.EXE to TENNIS so that we can modify it with DEBUG.
REN TENNIS.EXE TENNIS
- 3) Start up DEBUG and specify TENNIS as the file to modify.
DEBUG TENNIS
- 4) Search for the bytes to change.
S0 1000 E8 DD 84
- 5) DEBUG will respond with the address of the bytes to change in the form of:
XXXX:YYYY
- 6) Edit the provided address as follows:
E XXXX:YYYY B8 00 00
- 7) Write the software back out.

W

8) Exit DEBUG

Q

9) Rename TENNIS back to TENNIS.EXE
REN TENNIS TENNIS.EXE
That's all folks.

Marc sent more material than is printed here. I bumped the rest to issue #81. Sorry Marc. ...RDEXed one of the data epilogue bytes makes it impossible to catalogue a protected disk) that several of the files come up as

unClassifieds

How to place an UnClassified Ad

Send a typed sample copy with appropriate instructions. (If possible, send text on a 5.25" Apple format disk.) Use up to 40 characters per line, we will adjust word wrap.

Special Graphics Instructions: The first three words of the first line are printed in bold for free. If you want other words bolded, use 5 characters less per line. Use 10 characters less per line if you have a lot of uppercase bold letters. Bold letters are wider than normal. If the typed copy does not show bold, circle the words you want bolded and, on the side, write BOLD. If you want a line centered, write CENTER next to that line. There is no charge for centering any line.

You must check your ad for errors, the first time it runs. Errors on our part will be corrected, then, for free. Errors or changes on your part will be charged a \$5 processing fee.

★★★★ New Rates (per line) ★★★★★

Computist club member 25¢

All others 35¢

The minimum order is \$5.

• Our liability for errors or omissions is limited to the cost of the ad.

• We reserve the right to refuse any ad.

• Washington state residents add 7.8% sales tax.

• Send a check or money order (funds drawn on US bank only) for the entire amount to:

COMPUTIST unCLASSIFIEDS

33821 East Orville Road
Eatonville, WA 98328

BOOK and MAGAZINE SALE!!!

Back issue of 2600 (1985-90) \$2 each, A+ (1984-86,89) \$1, Apple Assembly Line (1980-88) \$2, Apple Orchard (1983-84) \$1, Apple Software Protection Digest v1 \$1, Bootlegger (v1-7) \$3, Byte (1977-91) \$2.50, Call APPLE (1982, 85-87) \$1.50, Circuit Cellar Ink (1989-90) \$1.75, COMPUTE! (1981, 85,86) \$1.25, COMPUTE! Apple Applications (1988) \$1.25, Computist (4-80) \$2.50, Creative Computing (1981-85) \$1, InCider (1983-88) \$1.25, Journal of Computer Game Design (1989-91) \$3.25, Micro Adventurer (1984-85) \$1, Nibble (1981-84) \$1.50, Open-Apple (1985-87) .75, Softalk (1981-84) \$1.50. Orders \$20 or over, take 25% off. \$50 or over take 50% off, except for AAL, 2600 & Computist, to which book discount applies.

Books:

Apple Reference Manual 1980 (worn) \$9
DOS 3.3 Reference Manual \$7
Apple Pascal Reference Manual \$18
Apple Pascal Operating Sys Manual \$8
Beneath Apple ProDOS \$15
Call APPLE: All About Pascal \$12⁹⁸
Peeking at Call APPLE v2 1979 \$13
Peeking at Call APPLE v3 1980 \$17
Best of Creative Computing v2 1976 \$12
Best of Creative Computing v3 1977 \$9
Best of MICRO v2 1978-79 \$9
Best of MIRCO v3 1979-80 \$11
Open Apple: vol 1 1985 \$9
Apple Machine Language, by Inman \$7
Software Author's Guide 1983 \$3⁸²
More BASIC Computer Games \$6
Apple II Users' Guide 1981 (385p) \$7
Apple Writers Minute Manual, II+ & IIe \$8
Proceedings of the 1990 & 1991 Computer Game Developers' Conference \$20
IBM VM/SP manuals, many titles inquire

On book orders of \$n, take (n/10)% off. Include 70¢ per magazine and \$2 per book for postage. I will refund excess or bill amount due. Minimum order of \$10.

Phil Goetz

4023 Huckleberry Row
Ellicott City, MD 21043

Wanted

Most Wanted List Software

? Need assistance to deprotect a disk?

Softkey hobbyist is interested in acquiring copy protected software to deprotect. Good track record, many successful attempts. Original disk will be returned along with Softkey for Computist. Especially interested in older software (pre-1988) but will give any disk a shot. System: Apple II+, 64K. Send disk to:

Rich Etarip

824 William Charles, Apt #2
Green Bay, WI 54304

TRADE YOUR APPLE SOFTWARE

Send your list of programs to trade. I have over 120 originals to trade.

Byron Blystone
PO Box 1313
Snohomish, WA 98290

BOOK SALE

Keys to Solving Computer Adventure Games. Maps, hints, clues, & some solutions. Black Cauldron, Kings Quest II, Leather Goddesses, & more. 26 games for Apple & others.

M.K. Simon 286 pgs
Prentice Hall \$15.00

Keys to Solving Computer Adventure Games, Book II. Maps, hints, clues, & some solutions. Indiana Jones, Gunslinger, Hollywood Hi-Jinx & more. 13 games for Apple & others.

M.K. Simon 292 pgs
Prentice Hall \$15.00

Add \$2.50 for postage & Insurance

Software — Books — Magazines

We buy & sell out-of-print & hard-to-find Apple II originals, old and new. Send \$1 for catalog.

Frank Polosky
PO Box 9542
Pgh, PA 15223

RDEX

Contributors:

Safaa Abdulla 15
Big Al 8
Marc Batchelor 22
John C. De La Cruz 19
William Degelmann 16
Norman Dodge 8
Bob Feigenblatt 8
David R. Hopkins 7
Jeff Hurlburt 4
Steven Kalynuik 21
Kilroy 16
Jay Kitaguchi 15
Krakowicz 17
Carl D. Purdy 16
Brian A. Troha 15
Terry Waskowich 8, 15
Don Westcott 21
Zorro 21

Most Wanted

80 Ace of Aces Accolade
65 Airheart Broderbund
63 Alcon Taito
74 Algebra Shop Scholastic
63 Alien Mind PBI Software
73 American History Explorer Series Mindscape
75 Anchorman Virginia Reel
74 Animals of the Past Focus Media
72 Ankh Datamost
73 Ant Farm Sunburst
67 Aquatron Sierra
69 Axis Assassin ?
63 Bad Street Brawler Mindscape
73 Bank Street Beginner's Filer Sunburst
73 Bank Street School Filer Sunburst
80 Battle Chess II Interplay
63 Beyond Zork Infocom
65 Bilestoad Datamost
69 Blue Powder - Grey Smoke Grade
74 Birds - Trees & Flowers Focus Media
63 Border Zone Infocom
65 Borg Sirius
67 Bouncing Kamungas Penguin
66 Boxing ?
65 Bureaucracy Infocom
67 C'est La Vie Adventure International
69 Caverns of Callisto Origin
69 Checker Odesta
69 Chess 7.0 Odesta
69 Chuck Yeager's Adv Flt Trainer Electronic Arts
75 Clue Master Detective Leisure Genius

80 Colony, The Mindscape
68 Comics Accolade
63 Cosmic Relief Datasoft
65 Crime & Punishment Imagic
69 Crossword Magic v4.0 ?
69 Cybernation Nexa Corp.
74 Decimal Dungeon Unicorn
74 Decisions Decisions: Colonization v1.0 Tom Snyder Productions
69 Delta Squadron Nexa Corp.
67 Desecration Mind Games
66 Disk Optimizer System Nibble Notch
65 Dondra Spectrum Holobyte
69 Dragon Eye Epyx
69 Dueling Digits Broderbund
68 D&D-Master Assistant vol2 SSI
62 DROL Broderbund
67 Epoch Sirius
74 Exploring Tables & Graphs Level 2 (SU) Weekly Reader
67 Evolution Sydney
67 Falcons Piccadilly
68 Factastics Trivia Daystar
75 Final Frontier Softsmith
73 Fisher's Cove Tom Snyder Productions
69 Fit Wars Sirius
74 Fraction Action Unicorn
69 Gemstone Healer SSI
73 Geometric Supposer (the) Sunburst
66 GEOS Berkley Softworks
72 Galactic Gladiators SSI
63 Gladiator Taito
73 Goodell Diamond Caper Tom Snyder Productions
67 Gorgon Sirius
66 GradeBuster 1 2 3 Grade Buster
61 Gutenberg Sr. Micromation LTD.
65 Halls of Montezuma Electronic Arts
80 Heat Wave Accolade
67 High Orbit Softsmith
67 Horizon V Softsmith
75 Hunt for Red October GS Datasoft
69 Impossible Mission Epyx
62 Indoor Sports Mindscape
68 Infocomics Infocom
66 Jane ?
63 Joker Poker Mindscape
72 Kabul Spy Sirius
71 Keyboarding Klass Mastery Development
75 King's Bounty New World Computing/Broderbund
68 Kingdom of Facts ... Santa Barbara/Thunder Mountain
75 Kobayashi Alternative (The) Simon & Schuster
72 Lane Mastodon Infocom
67 Lancaster SVS
72 Laser Force (Iigs) Britannica
75 L.A. Land Monopoly Softsmith
66 Legacy of the Ancients Electronic Arts
65 Lost Tomb Datasoft
74 Mammals - Reptiles & Amphibians Focus Media
65 Manhunter New York Iigs Sierra On Line
65 Mavis Beacon Teaches Typing (gs) Software Toolworks
73 McGraw-Hill Problem-Solving Lvl 5&6 Tom Snyder
74 Micro-Typewriter v3.1/4.0 S.E. Warner
67 Microwave Cavalier
66 Might and Magic II Activision
73 Mind Castle I MCE Inc.
69 Minotaur Sirius
63 Modem MGR MGR Software
68 Mr. Pixel's Cartoon Kit. Mindscape/Thunder Mountain
73 Mystery of Hotel Victoria Tom Snyder Productions
63 National Inspirer Tom Snyder Productions
75 Neptune Softsmith
66 Observatory (The) ... Mindscape/Lightspeed Software
74 Ocean Life Focus Media
66 Odin Odessta
63 Operation Wolf Taito
68 Pensate Datasoft/Softdisk
69 Phantasia II SSI
67 Phantoms 5 Sirius
67 Pig Pen Datamost
74 Plants & Animals of the Desert Focus Media
75 Prince of Persia (5.25") Broderbund
67 Project: Space Station Avantage
75 Promethean Prophecy (The) Simon & Schuster
67 Pulsar II Sirius
68 Pure Stat Basketball ?
62 Quadratic Equations II Olympus Educational Software
63 Questron II Electronic Arts

68 Rails West SSI
63 Renegade Taito
67 Rescue Raiders Sir Tech
67 Rings of Saturn - Level 10 ?
63 Rocket Ranger (Iigs) Cinemaware
69 Roundabout Datamost
75 Russki Duck Softsmith
63 S.D.I. (Iigs) Cinemaware
62 Sea Stalker Broderbund
67 Serpentine Broderbund
74 Seven Cities of Gold Electronic Arts
68 Skeletal System Brainbank
63 Sky Shark Taito
80 Sim City Maxis
63 Sound Song & Vision Advanced Software
67 Space Ark Datamost
62 Spare Change Broderbund
67 Spectre Datamost
62 Speedy Spides Readers Digest
67 Star Cruiser Sirius
67 Star Maze Sir Tech
63 StickyBear Math: Add & Subtract Optimum Resources
68 Stickybear GS Versions 3.5 Xerox
80 Stormovik Electronic Arts
67 Succession Piccadilly
65 Superstar Ice Hockey Mindscape
61 Superstar Indoor Sports Mindscape
74 Surveys Unlimited Mindscape
68 Talking Text Writer GS Scholastic
68 Tangled Tales Origin Systems
69 Tetris (Ile) Spectrum Holobyte
72 Theatre Europe PBI
74 The Other Side v2.0 Tom Snyder Productions
65 Thunder Chopper ?
63 Ticket to Washington D.C. Blue Lion Software
74 Time Explorers Gameco
74 Time Liner v1.1 Tom Snyder Productions
63 Tomahawk Electronic Arts
68 Tomahawk (Iigs) Datasoft
80 Tower Toppler U.S. Gold
69 Track Attack Broderbund
68 Triad Thunder Mountain
72 Triango (Iigs) California Dreams
68 Trinity Infocom
73 Unicorn 5.25" software Unicorn
73 Vincent's Museum Tom Snyder Productions
68 Volcanoes v1.8 Earthware Comp. Services
66 War in the Middle Earth Melbourne
80 Wayne Gretzky Hockey 2 Bethesda
67 Wayout Sirius
63 Wings of Fury Broderbund
63 Wizardry: Return of Werda Sir-Tech.
68 Word Attack Plus (Iigs) Davidson
65 Works (the) First Star Software
67 Zenith Softsmith

IBM Most Wanted

75 Empire Intersil
72 GBA Championship Football Electronic Arts
68 Graphitti George Best Phillips Academy
61 Gunship Microprose
63 Heros of the Lance SSI
72 Kings Quest III Sierra
72 Operation Wolf Taito
72 Radio Baseball Electronic Arts

72 Features, Notes and such: The Product Monitor • A Bug in Prentice Hall Science Courseware • A note on Stickybear encrypted sector • An accelerated IIE & EDD 4 • Finding the licensee's name in GEOS • A BUG in Teacher's Tool Kit Series • A reader review of the Trac Card • An Explanation of Self-sync Bytes • Another reason why Cookbook Cracks might not work • BBS News • Beginners Guide to "PACMAN" deprotection. • Bogus 18 sector disks? • Bug in Type! softkey • Bugs in Pool of Radiance • Changing levels on Tetris (IIE) • Cheats, Hints, and Tips for Neuromancer • Comments & possible help to other readers • Comments on IBM Ragging • Converting Print Magic Graphics to Publish It! • Copy Protecting Your Own Disks with ProDOS • Determining when individual files can be removed from a Protected Disk • DOS 3.3P (for protected) • Enhancing DOS 3.3 • FIND.CAT Enhancement • Half & quarter Tracks • Installing GS/OS on Sierra 3D Adventures • Lifting the Lid on COPYA • Making Jack Nicklaus Greatest 18 Holes of Championship Golf play faster • Moving Destroyer GS (Epyx) to Hard Disk • Note on '84 & '85 MECC disks • Notes on programs published by Micrograms • Notes on Time Out and Appleworks 2.1 • Notes on Ultima IV and V • Print in Color with Appleworks • Put Autoduel on a 3.5" diskette • Reading From Protected ProDOS Disks • Removing the Manual Check from Pirates! GS • Running other programs from your Hard Disk • Super 6.0 FastcopyA • Fun with Super 6.0 FastcopyA • Turn Dig Dug into a BRUNable File • Ultimapper V: a mapping program • Softkeys: Addition & Subtraction • Aesop's Fables IIE • Algebra 1 • Algebra 2 • Algebra 3 • Alphabet • An Introduction to General Chemistry • APBA Major League Players Baseball • Arkanoid • Arkanoid II: The Revenge of DOH (GS) • Balance • Balance of Power 1990 v. 2.08 • Biosolve • Bubble Ghost (GS) • Building memory Skills • Calendar Crafter v1.2 • CBS's Pathwords • Charlie Brown's 1,2,3's • Chessmaster 2100 v1.1 • Children's Writing & Publishing Center • Conquering Decimals * and / • Conquering Decimals + and - • Conquering Fractions; * and / • Conquering Fractions; + and - • Coordinate Math • D.C. Heath/MB • Dark Lord • DataEast Games • Decimal Concepts • Decimal Discovery • Decimals • Delta Drawing 3.33 • Designasaurus • Dig Dug • Duel (gs) • Early Games • Eliminator • Equation Math • Estimation • Explora-Classic series • Explora-Science Whales • Fraction Concepts • Fraction Practice Unlimited • Fraction Recogni-

tion • Fractions • Freddy's Puzzling Adventures • Frogger • Galaxy Math Games • Galaxy Search • Garfield Companion • Garfield Trivia • Geometry (GS) • Ghostbusters • Gnarly Golf (GS) • Gnee or Not Gnee • Grammar Examiner • Graphics Studio • Guinness World Records • Homeworkeer • Houghton Mifflin Math Courseware • Ice Demons • Invitation to Math series • Kid Niki • King of Chicago • Kittens, Kids, and a Frog • L.A. Crackdown • Language Carnival • Magic Slate II • Magic Word • Master Match • Math And Me • Math Blaster • Math Blaster Plus • Math for Everyday Living • Math in a Nutshell • Math Masters • Math Tutor (Percents • Mathematics Skills • Mavis Beacon Teaches Typing (IIE) • McGraw Hill Compucat Quizware • Memory Castle • Meteor Multiplication • Microzine #3 • Microzine #8 • Microzine #9 • Microzine #10 • Microzine #11 • Microzine #12 • Microzine #13 • Microzine #15 • Microzine #16 • Microzine #22 • Microzine #23 • Microzines and Microzine Jr. • Mixed Numbers • Moptown Hotel • Mr. and Mrs. Patatohead • Mr. Pixel's Programming Paint Set • Mystery Matter • Mystery Objects • Numbers • Opposite • Organic Chemistry • Pacman • Peanuts Math Matcher • Percentages • Perplexing Puzzles • Pick the Numbers • Picture Perfect • Pirates! GS • Pixelwerks • Platoon • Police Quest • Pool of Radiance • Quest Strategy Checker • Rainbow Painter • Ratios & Propotions • Read 'N Roll 1.1 • Read-Write-Publish • Reading Comprehension • Robomath • Sailing Through Story Problems • Scuffy and Friends • Sensible Speller (ProDOS) • Sierra Programs • Silicon Dreams • SOKO-BAN • Solving Quadratic Equations • Spelling Bee • Spelltronics • Spy's Adventure in South America • Square Pairs • Star Trek • Stickers • Study for Success • Success With Typing • Super Print • Tales of Fantasy • Talking Stickybear Opposites • Talking Text Writer • Teasers by Tobbs • Test Drive II: The Duel (IIGS) • Test Taking Made Easy • The Boars Store * • The Boars Tell Time • The Duel: Test Drive II • The Game Show • The Games: Winter Edition • The Wonderful World of Paws • Time Capsule • Times of Lore • Townbuilder • Transylvania • VCR Companion • Voyage of the Mimi • What makes a Dinosaur sore • Where in the USA is Carmen San Diego • Where in the World is Carmen San Diego? (GS) • Who, What, Where, When, Why • Whole Number Operations • WISCR Intelligence Test • Wood Car Rally • Word Attack Plus! Spanish • World Geograph • World History Adventure • Xenocide (GS) • Bitkeys: Borg • Great Western Shootout

• Laser Force • Math Blaster Mystery • The Hunt for Red October • APTs: Anti Gravity • Autoduel • Bard's Tale II • Chrono Warrior • Commando • Demonic Decks • Neuromancer • Pool of Radiance • Tetris • The Bards Tale III • Track & Field • Victory Road • Xevious • Zany Golf • Playing Tips: Bard's Tale II • Bard's Tale III • Defender • Defender Of The Crown (IIGS) • Leisure Suit Larry • Moebius • Pool of Radiance • Print Shop Companion • Robocop • Tower of Myraglen

• Wasteland • Who Framed Roger Rabbit • Wings Of Fury • IBM Softkeys: 688 Attack Sub • Battle Chess • Battle Hawks 1942 • Chuck Yeager's Advanced Flight Simulator v1.0 • Shinobi • IBM Feature & Notes: Bug in APT for Zany Golf • Cracking on the IBMpc •

and much more. . .

For a complete back issue list, send a 75¢ stamp to Computist.

Special Software Sale

Only \$10 each (while they last)

These software packages are NEW (shrink-wrapped except for the one copy of Sound Master that I opened in order to find out what it was). They're software packages that someone ordered and then canceled and we were unable to return.

SubLogic Scenery Disk 2 (Phoenix, Albuquerque & El Paso)

SubLogic

(All Apple II's) \$10.00

For use with Jet and/or Flight Simulator v2.0. Each scenery disk covers a geographical region of the country and includes major airports, radio-navigational aids, cities, highways, rivers and lakes located in that region. Enough detail is available for either visual or instrumental cross-country navigation.

Print Quick

Third Wave Technology Inc

(All Apple II's) \$10.00

Print Quick puts your most needed printer support programs on one disk. Quick Setup to put the control of all print features at your fingertips. Quick Typer turns your printer into an electronic typewriter by sending text right to the printer. Quick Dumper allows you to choose over 30 handy sign phrases or one button Hi-Res screen printing. Quick Font allows you to quickly change the shape and style of your printer characters. Text File Utilities will print text files or display them on screen with all the hidden command characters revealed. Owner's Manual, quick reference card, and detailed Help screens make this a useful print tool.

Victory Road, Ikari Warriors II Data East

(Apple IIe, c, gs, 128K required) \$10.00

Your heroes from Ikari Warriors are back. And this time, they're up against the monstrous Stonehead and his army of bizarre, inhuman creatures! Make your way through a desolate landscape, as hordes of bloodthirsty Green Grem-lins, Winged Man-beasts and enormous Worm-snakes threaten to overwhelm you. Do you have the courage to travel the Victory Road?

Murder Party

Electronic Arts

(Apple II+,e,c) \$10.00

Choose from two mysteries and personalize the characters to fit your players. Each time it's a different "party", so you won't be bored when you play again and again and again...

SoundQuest CZ Master

Sound Quest In

(Commodore Amiga) \$10.00

For use with the Casio CZ-101, CZ-1000, CZ-3000, CZ-5000 and other compatible synthesizers. Included are file management and bank editing features, patch mixing and random voice generation features. Compose and mix your own music using many of the package options available.

Send orders to Computist at the address listed on the Back issue order form below.

Issue	Mag Disk						
Core1	<input type="checkbox"/>	22	<input type="checkbox"/>	46	<input type="checkbox"/>	70	<input type="checkbox"/>
1	<input type="checkbox"/>	23	<input type="checkbox"/>	47	<input type="checkbox"/>	71	<input type="checkbox"/>
2	<input type="checkbox"/>	24	<input type="checkbox"/>	48	<input type="checkbox"/>	72	<input type="checkbox"/>
Core2	<input type="checkbox"/>	25	<input type="checkbox"/>	49	<input type="checkbox"/>	73	<input type="checkbox"/>
3	<input type="checkbox"/>	26	<input type="checkbox"/>	50	<input type="checkbox"/>	74	<input type="checkbox"/>
4	<input type="checkbox"/>	27	<input type="checkbox"/>	51	<input type="checkbox"/>	75	<input type="checkbox"/>
5	<input type="checkbox"/>	28*	<input type="checkbox"/>	52	<input type="checkbox"/>	76	<input type="checkbox"/>
6	<input type="checkbox"/>	29	<input type="checkbox"/>	53	<input type="checkbox"/>	77	<input type="checkbox"/>
Core3	<input type="checkbox"/>	30	<input type="checkbox"/>	54	<input type="checkbox"/>	78	<input type="checkbox"/>
7	<input type="checkbox"/>	31	<input type="checkbox"/>	55	<input type="checkbox"/>	79	<input type="checkbox"/>
8	<input type="checkbox"/>	32	<input type="checkbox"/>	56	<input type="checkbox"/>		
9	<input type="checkbox"/>	33	<input type="checkbox"/>	57	<input type="checkbox"/>		
10	<input type="checkbox"/>	34	<input type="checkbox"/>	58	<input type="checkbox"/>		
11	<input type="checkbox"/>	35	<input type="checkbox"/>	59	<input type="checkbox"/>		
12	<input type="checkbox"/>	36	<input type="checkbox"/>	60	<input type="checkbox"/>		
13	<input type="checkbox"/>	37	<input type="checkbox"/>	61	<input type="checkbox"/>		
14	<input type="checkbox"/>	38	<input type="checkbox"/>	62	<input type="checkbox"/>		
15	<input type="checkbox"/>	39	<input type="checkbox"/>	63	<input type="checkbox"/>		
16*	<input type="checkbox"/>	40	<input type="checkbox"/>	64	<input type="checkbox"/>		
17	<input type="checkbox"/>	41	<input type="checkbox"/>	65	<input type="checkbox"/>		
18	<input type="checkbox"/>	42	<input type="checkbox"/>	66*	<input type="checkbox"/>		
19*	<input type="checkbox"/>	43	<input type="checkbox"/>	67	<input type="checkbox"/>		
20	<input type="checkbox"/>	44	<input type="checkbox"/>	68	<input type="checkbox"/>		
21	<input type="checkbox"/>	45	<input type="checkbox"/>	69	<input type="checkbox"/>		

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

- ☆ Limited supply — first-come-first-serve basis.
- Out-of-print — only "Zeroxed" copies for sale.
- * Issue 66 is laser printed on 8 1/2 by 11 paper.

Back Issue Order Form

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

Back Issue and Library Disk Rates

	Quantity	US, Canada & Mexico	All others
Back issues	5 or less	\$4.75	\$8.75
	6 to 9	\$3.75	\$6.00
	10 or more	\$3.00	\$5.00
Zox* back issues	any qty.	\$4.75	\$8.75
Library disks	5 or less	\$5.50	\$7.50
	6 to 9	\$4.00	\$6.00
	10 or more	\$3.00	\$5.00

*Due to the time and effort involved in making Zox copies, their price will remain at \$4.75 each for US, Canada & Mexico and at \$8.75 for all other Foreign.

Shipping is included in all the prices shown.

What's a library disk?

A library disk is a 5 1/4 inch floppy 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.

For a complete back issue list, send a 75¢ stamp to Computist.

Number of back issues. \$ _____

Number of Zox back issues. \$ _____

Number of library Disks. \$ _____

Washington state residents add 7.8% tax \$ _____

Total enclosed \$ _____

Name _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

VISA _____

MC _____ Exp. _____

Signature _____

• US funds drawn on US bank. • Most orders shipped within 5 working days, however please allow up to 4 weeks delivery for some orders. • Large orders are shipped UPS so please use a street address. • Offer good while supply lasts. • Call (206) 832-3055 to use a credit card or send check/money order to:

COMPUTIST 33821 E Orville Road Eatonville WA 98328