77 COMPUTIST

Canada & Mexico $7

U.S. $3.75

Table of Contents

COMPUTIST
33821 Orville Rd E
Eatonville WA 98328

ADDRESS CORRECTION REQUESTED

COMPUTIST

Charles R. Haight — **Editor**
Karen Fitzpatrick — **Circulation**
Jeff Hurlburt — **Reviews**
Dave Goforth — **BBS**

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

COMPUTIST
33821 East Orville Road
Eatonville, WA 98328
(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.

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 recieve 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 a 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.

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 **openapple 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:

```
10HOME:REMCLEAR 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 (◊) charactors. **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: **R**eader's **D**ata **EX**change. 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 relevent 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.

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 preceeded 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** .

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

Table of Contents

Ed Lee — TX

Advanced Playing Technique for...

Warlock GS

Three Sixty

Here is a revision to the Warlock APT in issue #73 by Don Westcott. I tried to use Don's APT, but the bytes were not at the right locations on my version. So I whipped out the old trusty Copy II+ and scanned for bytes ED 23 01 8D as Don had on his APT. When the search didn't find them, I thought, it's a ProDOS 16 program, why not look for it with relocatable code. Sure enough, when I scanned for bytes ED 00 00 8D it showed up 3 times. Once I changed the bytes it worked! So here goes, make these edits (on a backup disk of course.)

Blk	Byte	From	To
$240	$1D1	ED 00 00 8D	AD 00 00 AD
$241	$068	ED 00 00 8D	AD 00 00 AD
	$0C6	ED 00 00 8D	AD 00 00 AD

Just goes to show, that you should send all information you can on APTs or softkeys. Thanks again to Don for this APT.

To Rob Fiduccia: On his question in issue 73 about searching memory with the monitor on the IIgs (best thing around to do APTs with). Here goes:

\ bytes \< start addr . end addr **P**

This will allow you to search an address range for a sequence of bytes. For example: to search for bytes 00 67 B1 at address range $0000 to $FFFF you would type (while in the monitor):

\00 67 B1\<0000.FFFFP

If the bytes are found, the address location of every occurrence will be displayed on the screen. So have pen and paper ready, they scroll by real fast.

Here are some more useful monitor commands I have discovered. The list may not be complete if someone has more, please send them in for the rest of us to see, also is there is a reference manual or anything in using the monitor and where can I get a hold of one? And what function does the Memory Peeker CDA serve?

IIgs monitor commands (partial list):

ctrl E Displays all registers and values in them. (I don't know how to change them)
ctrl B Cold starts into BASIC (you lose any basic program in memory.)
ctrl C Warm starts into BASIC (you keep any basic program in memory.)
ctrl R Restores registers to startup values (OK,.. so I can reset them)
ctrl T Sets screen to text page one (useful for program crashes)
N Sets screen to normal mode
I Sets screen to inverse mode
! Goes into the mini assembler
= *decimal value* Decimal to Hex conversion (very useful)
***hex value* =** Hex to decimal conversion (bank/address)
***addr*L** Disassemble code starting at address
bank/address . bank/address Displays range of memory
esc 8 Sets screen to 80 column mode, also works while in BASIC
esc 4 Sets screen to 40 column mode, also works while in BASIC

Playing Tip for...

Ancient Land of Y's

?

Last but not least is a playing tip for Ancient Land of Y's using the handy dandy Visit Monitor CDA. (I'd rather use the monitor than editing the save disk as this will save time. I can edit and go right back into the game instead of having to reboot.) Enable your CDA for Visit Monitor. The character is located in memory locations 01/71E4 to 01/71F4? This is a breakdown of what I've found so far.

01/71E4 Char Hit Point (max $FF)
01/71E6 Char Hit Point minus damage
01/71E8-9 Experience points
01/71EE-F Money
01/71F2 Strength

Changing your experience points seems to be the best bet, since when you move up a level your hit points and strength are reset to what they should be plus a gain for level advancement. The maximum experience points is FF FF which is 65535 in decimal. This will give you 255 hit points and a level of 7. I haven't found the location for weapons or armor, but increasing experience points and boosting money seems to work for me.

Ⓟ After playing the game for a while I am now stuck in the mines (try to map realistic caverns on graph paper) and if anyone knows how to get through, your experiences would be greatly appreciated.

Well, that's it. Hope this helps someone. Keep on computing.

Edison — NE

BUG in Super 6.0 Fastcopy

There are two errors in the program SUPER 6.0 FASTCOPY A both on the disk and in the listing of issue #72 page 21. They are corrected here.

```
500 VT=9: FOR I=7 TO 12
530 IF CH$ = "N" AND I=12 THEN 580
```

Softkey for...

Double Check

?

Upon booting this disk you see the "]" meaning, hopefully, a near normal DOS. I tried to copy the disk with Locksmith Fastcopy and got all errors. This tells me to suspect everything. Looking at the disk with a nibble editor, (CopyII+, The Nibbler) you will find the normal bytes are changed from DE AA to AA AA on both the Address and Data epilogs. Very easy to correct but which method should I use? Try any of these.

The COPYA method

The COPYA method may or may not produce errors on the duplicate.

1. Boot a DOS 3.3 system disk.
2. Tell DOS to ignore checksum and epilog errors and use COPYA to copy the disk.

POKE 47426,24
RUN COPYA

Using SUPER 6.0 FASTCOPY

Select (L)ocksmith

L
1 *read bytes*
Y *Alter Address bytes*
N
N
N
AA *Change from DE*
N
Y *Alter Data bytes*
N
N
N
AA *Change from DE*
N
N
4 *Enter LS Fastcopy*

Super IOB method

Merge the following controller with SUPER IOB and run it.

Controller

```
1000 REM DOUBLE CHECK.CON
1010 TK = 0:ST = 0:LT = 35:CD = WR
1020 T1 = TK: GOSUB 490: RESTORE :
  GOSUB 170
1030 GOSUB 430: GOSUB 100:ST = ST
  + 1: IF ST < DOS THEN 1030
1040 IF BF THEN 1060
1050 ST = 0:TK = TK + 1: IF TK <
  LT THEN 1030
1060 GOSUB 230: GOSUB 490:TK =
  T1:ST = 0
1070 GOSUB 430: GOSUB 100:ST = ST
  + 1: IF ST < DOS THEN 1070
1080 ST = 0:TK = TK + 1: IF BF = 0
  AND TK < LT THEN 1070
1090 IF TK < LT THEN 1020
1100 HOME : PRINT "DONE◊WITH◊COPY"
  : END
5000 DATA 170,170,170,170
```

Checksums

```
1000-$356B  1040-$51E4  1080-$BA6E
1010-$3266  1050-$FDC1  1090-$CD24
1020-$7F0A  1060-$4016  1100-$83FF
1030-$690B  1070-$4813  5000-$6139
```

The edits are not in this controller because I didn't think you would want to type 78 Data statements.

Boot a sector editor and change all these locations.

Trk	Sct	Byte	From	To
$03	$06	$CB	31 37 30	32 32 32
		$D6	31 37 30	32 32 32
$07	$02	$60	31 37 30	32 32 32
		$6B	31 37 30	32 32 32
$07	$03	$18	31 37 30	32 32 32
		$23	31 37 30	32 32 32
$12	$08	$BD	31 37 30	32 32 32
		$C8	31 37 30	32 32 32
$12	$0C	$26	31 37 30	32 32 32
		$31	31 37 30	32 32 32
$13	$0B	$33	31 37 30	32 32 32
		$3E	31 37 30	32 32 32
$13	$0E	$4A	31 37 30	32 32 32
		$55	31 37 30	32 32 32
		$DF	31 37 30	32 32 32
		$EA	31 37 30	32 32 32
$18	$0B	$C5	31 37 30	32 32 32
		$D0	31 37 30	32 32 32
$1B	$01	$39	31 37 30	32 32 32
		$44	31 37 30	32 32 32
		$4F	31 37 30	32 32 32
		$5A	31 37 30	32 32 32
$1E	$0C	$D0	31 37 30	32 32 32
		$DB	31 37 30	32 32 32
$21	$0C	$F2	31 37 30	32 32 32
		$FD	31 37 30	32 32 32

After converting this disk to normal format AND editing it, you may copy the files to a fast DOS that has the VERIFY command intact. These programs change control from one another by BLOADing code into memory then changing the Applesoft start of program pointer to the new code, leaving all variables intact. I think it is a very nifty trick. It also made it VERY difficult to change the POKEs that mess up DOS when its looking for the original.

Menu Program

When I booted the disk for issue #72, I noticed a small seven sector menu program that does nothing but run a choice from the Catalog. I thought to myself what a waste of space, I have been using a three sector menu program for several years that does that! Wondering if someone else could use it, I decided to send it in. Then I thought to myself 'this could be a lot better', so I rewrote it and now it loads, locks, unlocks, deletes, and renames files also. It still only takes five sectors, plus it is easily modified.

Line 100 - CATALOG the disk.
Line 110 & 120 - print the letters and brackets then set the string that prints at the bottom.
Line 130 - print the string while waiting for a key press.
Line 140 - set command, if not a number then 220.
Lines 150-200 - reset command.
Line 210 - get letter of program.
Line 220 - check if letter is on the menu.
Line230 - check if program is binary.
Line 240 - check if text.
Line 250 - read name of file and add to command string, if rename command then 270.
Line 260 - print command then do it, END.
Line 270 - get new name for file then 260.
Line 280 - subroutine that checks the screen.

Menu

```
100 CLEAR : SPEED= 255: NORMAL :
  HOME :D$ = CHR$ (4): PRINT D$
  "CATALOG"
110 T = 0:CH = 4: FOR CV = 0 TO
  23: GOSUB 280: IF C > 175 AND C
  < 186 THEN POKE P - 1,219: POKE
  P,T + 193: POKE P + 1,221:T = T
  + 1:S = CV
120 NEXT CV: VTAB 24:A$ = "TYPE◊
  LETTER◊TO◊RUN,◊OR◊LOAD=1◊LOCK=2◊
  UNLOCK=3◊DELETE=4◊RENAME=5◊EXIT
  =6...."
130 HTAB 1: PRINT LEFT$
  (A$,39);:A$ = MID$ (A$,2) +
  LEFT$ (A$,1):K = PEEK ( -
  16384): IF K < 128 THEN FOR K =
  1 TO 75: NEXT K:K = FRE (0):
  GOTO 130
140 TEXT :B$ = "RUN" : POKE -
  16368,0:K = K - 176: IF K < 1
  OR K > 6 THEN 220
150 HTAB 1: CALL - 868: IF K = 6
  THEN END
160 PRINT "◊PRESS◊'LETTER'◊YOU◊WISH
  ◊TO◊" ;: IF K = 1 THEN B$ =
  "LOAD"
170 IF K = 2 THEN B$ = "LOCK"
180 IF K = 3 THEN B$ = "UNLOCK"
190 IF K = 4 THEN B$ = "DELETE" :
  FLASH
200 IF K = 5 THEN B$ = "RENAME"
210 PRINT B$;: CALL - 198: NORMAL
  : GET K$:K = ASC (K$) - 48
220 IF K < 17 OR K > T + 16 THEN
  130
230 CH = 1:CV = S - T + K - 16:
  GOSUB 280: IF C = 194 AND (B$ =
  "LOAD" OR B$ = "RUN" ) THEN B$
  = "B" + B$
240 IF C = 212 AND B$ = "RUN" THEN
  B$ = "EXEC"
250 FOR CH = 6 TO 39: GOSUB 280:B$
  = B$ + CHR$ (C): NEXT CH: IF GE
  = 1 THEN 270
260 HTAB 1: CALL - 868: PRINT B$:
  PRINT D$;B$: END
270 B$ = B$ + "," : PRINT : INPUT
  "WHAT◊DO◊YOU◊WANT◊THIS◊PROGRAM◊
  RENAMED◊TO" ;T$:B$ = B$ + T$:
  GOTO 260
280 C1 = INT (CV / 8):C2 = CV - C1
  * 8:P = 1024 + 128 * C2 + 40 *
  C1 + CH:C = PEEK (P): RETURN
```

Checksums

```
100-$7F0F  170-$0EA1  240-$589B
110-$606F  180-$523B  250-$E247
120-$2607  190-$2F05  260-$6B89
130-$8907  200-$F3F8  270-$2FFC
140-$3A84  210-$4C72  280-$1A23
150-$1A3D  220-$1B18
160-$C89D  230-$45DC
```

I use this almost exclusively. The only thing I didn't really like about the program was that it shows a standard catalog of all files. I used to rename secondary files 'A' with several **ctrl H**s to hide the names, but the only way to access these was with special loader programs. Now thanks to Zorro in #72, all I have to do is Unlock the files I don't want showing and add these two lines to menu for locked files only.

Patch DOS to not show unlocked files.

```
95 POKE 44452,255:POKE 44513,67
```

Fix DOS.

```
105 POKE 44452,22:POKE 44513,2
```

Add this if you have a Pronto-Dos that uses the TYPE command.

```
245 IF C=212 AND B$="LOAD" THEN
  B$="TYPE"
```

Otherwords

I for one am sick of looking at that LONG Most Wanted List. It seems most have been covered before in back issues and in the books of softkeys. Please look in the back issues first! For those who don't know, you may get a complete back-issue list from Computist for the amazing low price of one self addressed 9 x 12" envelope with $.45 postage on it.

The easiest way is to send a 65¢ stamp (only) and ask for the back issue list. No point in paying the postage for a 9" x 12" envelope twice. ..... RDEXed

Help needed

Ⓟ I need help with Clue Master Detective v1.5 by Leisure Games. Here's how far I got. Booting this disk you see it is ProDOS based. So I tried to copy the disk with Locksmith Fastcopy and got Data errors. This tells me the address headers (epilogs) and trailers (prologs) are OK. That leaves the Data epilogs and/or data check sums and/or prologs at fault. Looking at the disk with a nibble editor, (CopyII+, The Nibbler) you will find the Data epilog bytes are at fault. The normal bytes are changed from DE AA EB to AA DE EB. Through trial and

error I found out this disk MUST be formatted with a volume number of one. (Is that a ProDOS limitation?) Boot a sector editor and change this byte in the ProDOS file so it will load from the disk. Change track $21, sector $07, byte $C8 from AA to DE. Here's where I need help. The copy will 'randomly' freeze the game or select two rooms instead of a room and a weapon. There is some code on track one that I think is a signature check. It starts like this...

```
AD 30 BF    LDA  $BF30   |
29 7F       AND  #$7F    |-- get slot
AA          TAX          |
BD 89 C0    LDA  $C089,X  turn on drive
```

More help needed

I have decided that I am tired of looking at my stack of unused software. I am in serious need of clues, hints, patches, and outright answers to several computer video games and almost all of my text adventure games.

A good example would be Zorro by Datasoft 1985. I am confident that I have mastered the game. My high score is over 78,000 but I still can't finish the game! How do I get the rose? Here is what I have collected, in order- handkerchief, boulder to last pully, wine glass, horseshoe, 2 bells, 6 bags of money, and the boot. What have I missed?

Dark Lord by Datasoft 1987 has defied completion for over two years by me, a friend with an apple, and my cousin to whom I must loan my computer. How do I get past the Guard? Is there another way in?

I purchased the Zork Trilogy because I liked the Eamon text adventures. After playing Zork I for a few weeks I was extremely frustrated so I purchased Zork I with on screen hints. I have cheated and read all the clues so I am sure I have mapped everything and been everywhere but I still haven't completed the game. That #@*&% thief keeps stealing the torch then my lantern goes out.

Deciding Zork was too hard I purchased Wishbringer by Infocom 1985. It says 'Introductory Level' but I can't seem to get any where. I have gotten back across the bridge but I am lost from there on. What am I supposed to be doing?

I like to consider myself of above average intelligence and I love the challenge of mapping these games, but my skills as an adventurer are nil. It should be noted that I almost always win at Conan, Karateka, 221 B Baker St., Clue Master Detective, and Wheel of Fortune. I had an almost complete map for Below the Root before it blew up. A lot of my software is old so clue books are hard to find.

I would like to see enough interest generated from my tales of frustration to add a Questions and Answers section to Computist. Maybe a half a page or so, not too big, just enough to help all of us who are quietly pulling our hair out. I would love to hear from people who have completed these games, either directly or through Computist. Here are a few of the games I need help with. Ultima I - III, Zork I - III, Wizardry Proving Grounds, Gemstone Warrior, Plundered Hearts

Ephraim Santiago — IL

Softkey for...

Moptown Hotel v1.3

The Learning Company

Requirements:
The original Moptown Hotel disk (5.25")
1 blank disk
COPYA
Copy II Plus
Any disk with normal DOS 3.3

This softkey would not be possible if it weren't for the softkey from Jack R. Nissel on Reader Rabbit in issue #68, page 22. I've tried all those other softkeys in previous issues and got nowhere with those complicated methods.

1. Boot your DOS 3.3, system master disk and at the "]" prompt enter.

POKE 47426,24 *ignore checksums and epilogues*
RUN COPYA

2. Put your original away and boot Copy II Plus. Use the COPY DOS function to copy normal DOS 3.3 from any disk to your copy, (target disk).
3. Now select the Change Boot Program function from Copy II Plus to make sure that the Boot Program name is HELLO. If it is not, then use this function to change it to HELLO.

The Last Print

I've been a subscriber for about a year now and I've been reading all these negative comments about the way Computist is printed or how the price is outrageous. Let's not mention any names, you know who you are and your complaints.

I appreciate the hard work that's put into this paper and the information. After all if it weren't for Computist where else could we look forward to for Apple IIe, IIgs, Mac, and IBM and its programs in terms of backing up your software and the vast amount of technical information.

Do you ever stop and think what would you do without Computist or if it just went under because of lost subscribers or financial reasons, or maybe the volunteers who donate their time and effort so that we could just sit back and wait for that next great issue of Computist that we take for granted. I think it's that time again where we should be asking ourselves how can we help to insure the survival of this subscriber supported magazine, tabloid, or call it what you like, but one thing's for sure "When it's gone, it's gone!"

Softkey for...

Animal Kingdom

Unicorn Software

Requirements:
DOS 3.3
Copy II+
One blank disk

The credit goes out to Spike. In issue #69, Spike informs us on how to backup All about America for the Apple IIe 5.25" disk.

Animal Kingdom uses only one disk which is doubled sided. To make a back up use the following steps.

Remember: ALWAYS write protect your original disk.

1. Boot your DOS 3.3 system disk.
2. Tell DOS to ignore checksum and epilog errors and use COPYA to copy both sides.

POKE 47426,24
RUN COPYA

3. Boot Copy II+ and delete the DOS image from your target disk. (Not your original disk)
4. Copy DOS from your DOS 3.3 system disk onto your target disk.
5. Rename the file "Jello" (on your target disk) to "Hello". That's it!

You know have a deprotected backup. Have fun!

The CPR Agent — Canada

Softkey for...

French Grammar I-VII

Queue/Intellectual Software

These programs are designed to review the basic grammar structures of the French that is studied in schools. There are ten lessons on each disk which includes a review lesson. The form of the program is very standard and has a no-bells-and-whistles approach. A teacher management module can be purchased separately (for $10 a disk) but you must send your disk back to have it installed. Luckily one of the disks in the set that I worked with had the management unit installed. It took no time at all to figure out that the other disks were only missing a couple of files. After transferring these files to the other disks, they too had a teacher management module to keep track of the averages of the students who used the disk — and it didn't cost $10 a disk!

The de-protection of these disks (and all Queue or Intellectual Software disks that I have worked with) is very standard and, in fact, can be used on all the disks.

1. Capture the RWTS from one of the disks using your favorite technique. (If you have trouble getting this, find a friend or neighbor who has a Laser computer. The CONTROL-RESET M technique for getting into the monitor on these machines, works marvelously well!) Keep this RWTS around as it seems to work on everything Queue puts out!
2. Once you have the RWTS, simply use it with Super IOB and one of the SWAP CONTROLLERS to copy the disk.
3. Use Copy II Plus (or a similar utility) to change the boot file from "HELLO" to "BTUT" (or the first file in the list - it was BTUT on every disk that I worked with in this series.)

The de-protection is done — unless you want that teacher management module on each of your disks. To get it, you first have to have it already installed on one of your programs or one of your friends programs. Simply copy the TUT file and the ACSES file (both Applesoft) from the disk with the teacher management module to your disk. You will find that your copy program will tell you that the TUT file already exits on your disk. Copy over it with the new TUT file. Your original TUT file occupies only 50 sectors while the one with the management module covers 59. The code for accessing this management area is **ctrl M** at any time the program is not asking for a specific answer to a question. I should say, it is **ctrl M** for all Apple II's and II+'s. For the Apple IIe's, IIc's, and the Laser computers, it's **ctrl]**. After copying the files, it would be a good idea to check it out to make sure it works.

One word of caution. Be careful with the lesson files on the disk as they all have control characters embedded in their names. There is really no reason to touch them unless you are that enterprising teacher out to add a little character and flavor to these otherwise rather drab French exercises! Finally, be sure to use a fast DOS — or patch DOS 3.3 with any one of several fast patches. Diversi DOS is a good choice since it handles text files particularly well and seems to be very compatible with most programs. The fast patch that comes as part of Advanced Ideas' Ultra Disk Pak works very well too.

Softkey for...

Alphabet Circus

DLM/Neosoft

This is an excellent program for teaching pre-schoolers and first grade students the alphabet. There are six different programs on the disk to keep the learning activity varied and the motivation high.

Requirements:
fast copier (I used Copy II Plus)
sector editor (I used Copy II Plus)

1. Fast copy the disk (there are no errors or glitches).
2. Sector edit:

Trk	Sct	Byte	From	To
$06	$0D	$5F-60	A9 56	60 60

Or search for BD 89 C0 A9 56. It can be found in the file HELLO3.

Details

This program uses one of the shorter DOS's and then fills the entire disk with program. Consequently, if you try to transfer all the files to another disk, you get the joyous DISK FULL message before you finish the file transfer. After several attempts to make enough space to actually transfer all the files, (and failing!) I discovered the best thing to do was to leave it all on the fast copied disk and search for the signature check.

I decided to begin this process by examining the files on the disk, only to discover that track $11 was empty! I found the catalog on Track $03 and moved a copy of it sector by sector back to track $11 where it ought to be. (This is actually a pointless exercise since the program could care less what is on track $11 — it looks to Track $03 for its directions. But seeing the catalog on Track $11 makes ME feel much better about everything!)

In looking over the catalog I noticed three Hello files — HELLO, GAME. HELLO, and HELLO3. The first two are Applesoft BASIC, easily loaded and listed. Nothing seemed unusual in either. The third one, HELLO3, was a binary file occupying 2 sectors (one sector actually, since the first simply pointed to the second). I was immediately suspicious. I disassembled and there it was, a nice little signature check for several non-standard bytes. I placed two 60's right after the instruction to turn the disk drive on and gave it a try. It loaded and ran more smoothly than the original. QED!

Softkey for...

Early Games - Music for Children

Counterpoint Software

Requirements:
FID (from the DOS Master)
an initialized disk (preferably with a fast DOS)
Copy II Plus (or similar utility for changing the boot file on a DOS disk)

This is an interesting piece of software that allows children to develop short musical routines in the bass or treble clef, save them, and play them back. It also has a routine for putting kaleidoscopic graphics to the music — something children would very much enjoy. Finally it has the facility to quiz children on musical notes, sounds, etc. Of course, the real problem with this program is that it is copy protected. Once again we have a potentially excellent program delicately balanced under the umbrella of a copy protection scheme that will crash the whole disk the minute a youngster (for whom the program was designed) does something out of the ordinary that the computer doesn't expect. A very intelligent move, Counterpoint!

Step-by-step

1. Boot your DOS Master and.

CALL -151 *enter the monitor*
B954:29 00 *ignore 1st byte of Addr Prologue*
BRUN FID

2. Use FID to copy the files. Follow the prompts (Use the wildcard "=" for the file name and answer "N" to the "Do you want to be prompted?" question).
3. Use Copy II Plus (or similar facility) to rename the boot file the first file on the list.

Details

There are really only two or three potential problems to de-protecting this disk. I used FID rather than COPYA because track $04 was all FF's and the first three tracks had an altered Address Epilogue. Rather than have my disk drive "rattle" through track $04 and rather than try to get the somewhat delicate balance of ignoring prologue AND epilogue changes, it seemed a better idea to go with FID. However, it was a bit of a surprise to watch FID copy the first several "blank" files! FID seemed to be copying files that had no name! This, I remembered, was a sure sign of files that were masked by control characters. The best way to handle these files is to NOT get into a position where you have to enter the name of the file. Hence, let Copy II Plus (or other similar utility) identify the first file as the new boot file. After that final change, the disk works fine.

Softkey for...

French Word Games

EMC

Requirements:
COPYA
DOS System Master disk
Copy II Plus

This is another of those review programs that helps get away from the classroom

routine and yet provides some very valuable instruction. There are three games on the disk — students can play by themselves or against a partner (a good choice if there are not enough computers to go around!). The games include matching, choosing the one that does not fit, and opposites. About 1500 words are taught and no specialized computer skills are required.

Step-by-step

1. Boot your DOS 3.3 system disk.
2. Remove the master disk and insert a blank disk in the drive. Initialize a disk with HELLO1. Set the disk aside.

```
INIT HELLO!
```

3. Tell DOS to ignore address epilogs and use COPYA to copy the disk.

```
POKE 47503,14
RUN COPYA
```

4. Use Copy II Plus, FID, or similar utility to transfer all the files from the COPYA disk to the initialized disk.

Details

A fast DOS is not necessary as this program has 2 short routines that it loads immediately upon booting that put regular DOS 3.3 into hyper-drive! Perhaps they could be used in other places on other disks?!! They are called SPEEDOS1.OBJ and SPEEDOS2 .OBJ and each is a binary file of 2 sectors.

Softkey for...

Garfield, Eat Your Words

Random House

Requirements:
COPYA
sector editor (I used Copy II Plus)

Step-by-step

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

```
POKE 47426,24
RUN COPYA
```

3. Make the following sector edits to the copy.

Trk	Sct	Byte	From	To
$00	$03	$91	AA	DE
		$9B	DE	AA
	$06	$AE	AA	DE
		$B3	DE	AA

That's it!

Notes on Les Secrets de Gertrude

Two of the programs that caused me the greatest consternation in de-protection were Gertrude's Secrets and Gertrude's Puzzles, both by the Learning Company. The reason they were such a challenge was because they both used half tracks as part of the encryption - at least the copies that I had both used half tracks. I spent many hours devising ways to access the data from the originals and getting it back on to a standard disk. All to no avail. Since Copy II Plus would back the disk up, I was in no rush to have a deprotected copy. But like a tiny rock in your shoe, it is always a bit of a pain.

So when our media center acquired a copy of Les Secrets de Gertrude (Gertrude's Secrets in French!), some lights came on and the laboratory of my mind began conjuring a scheme! The big bonus with the French program was that I noticed the distributor was Gessler - a company that I had encountered before and had little trouble with. The plan that chased around in my mind as I drove the disk home was to rid the disk of Gessler's protection technique, call up the sectors that had the instructions with a sector editor, change (translate) the instructions from French back to English, write it all back to the disk, and PRESTO an English version of the program without the pain of half tracks! Since I could handle myself reasonably well with the written form of the French language, I was confident that with a little luck I could perform one of the ultimate sneaky hacking tricks.

As I examined the disk there were, as you may have guessed, a number of surprises. This time, however, they were all in my favor. Surprise number one was that the disk was not copy protected at all! I was elated. I checked and rechecked as I could not, at first, believe it. The second surprise came as I located the French instructions using the sector editor. I began carefully substituting words and letters, hoping that I wouldn't mess up some call routine by not enough of one or too many of the other. During this process, I began searching for the proper value to enter for an "x" and not finding it on the sector where I was presently working, I wrote what I had changed to the disk and began backing up through other sectors, looking for an "x" somewhere.

Lo and behold I ran into a sector in English! I stared in disbelief again! What recklessness was this? I determined that the program probably had some key press that you could enter in the middle of whatever you were doing, and you could "help" yourself with your problem by having an instant English translation. It was a good idea but I was wrong. As it turned out, the whole of that particular file was English. I asked an immediate question (out loud, I might add) "How many other files do we have on this disk that are in English?" After several minutes of careful rummaging through files and sectors the answer turned out to be "All the necessary files (including the HELLO file) for an English version of this program are already here on the disk!!" I was stunned! I have no idea why a software company would do this but I am most MOST grateful to them!

What is to be learned from all this? Two things. First and most important is that sometimes the answer to our immediate problem is so obvious that it is overlooked. (I may have spent hours and hours translating the instructions and never ever even realized that the solution was next door in the previous sector!) Secondly, there may be more information on an innocent disk than we even realize. Since this particular program is fairly short, there is room on the disk for both versions and then some. Finally, it is sometimes interesting and very worthwhile to check a de-protected disk for deleted files. On one of my programs I found three deleted files that proved to be evolutionary stages of growth for one of the actual functional files in the program. It was an education in programming to follow the "thinking" of the programmer as he developed the file.

Now for Gertrude's best guarded secret. If you have the program Les Secrets de Gertrude in your library, here's how to have two programs for the price of one!

1. Boot a DOS 3.3 disk. Insert a blank disk and initialize it (with a fast DOS) so that it will BRUN FRESHSTART.O as its Hello file.

```
POKE 40514,52          to let the hello be binary
INIT FRESHSTART.O
DELETE FRESHSTART.O
```

2. Copy the following files from your original disk to the copy.

```
FRESHSTART.O
SECRETSHELLO.O
SECRETLOGO
OLD SECRET WORLD.IIE
OLD SECRET WORLD
SECRETCODE.O
SECRETSGOOSE.O
SECRETSRANDOM.O
OLD SECRETFONT
ADV #108
```

3. Rename the following files:

from	to
OLD SECRET WORLD.IIE	SECRET WORLD.IIE
OLD SECRET WORLD	SECRET WORLD
OLD SECRETFONT	SECRETFONT

Please note that it is the letter "O" that follows the period in several of the files and NOT the number "0".

That's it! Whoever would have guessed that the French were the guardians of Gertrude's best kept secret?! Viva le hacking!

Softkey for...

Chariots, Cougars, and Kings Kittens, Kids, and a Frog

Hartley

Requirements:
COPYA (or COPYA+)
sector editor

These are two fairly good programs that address reading comprehension in the primary grades. They are essentially stories (short) that the student is required to read and then there are a number of questions that the student is required to answer. Some educators would say that you could do just as well on paper with a lot less hassle. Others would say that the immediate feedback offered by the computer to an incorrect answer is much better than the red X — and the computer keeps track of it all. Further, there are some (a few) excellent graphics that go with the program. The biggest problem is that they are both copy protected which means that the disk will go down in no time at the hands of the "little people." So we shall do away with that problem.

Step-by-step

1. Use your System Master, your favorite method, or whatever to get to COPYA. (Note: use COPYA+ to achieve the same results as below if you wish.)

```
RUN COPYA
ctrl C                  At the first prompt
70                      to prevent reloading COPY.OBJ
CALL-151                enter the monitor
B925:18 60              to ignore data epilogues
B988:18 60              to ignore address epilogues
BE48:18                 to ignore errors
ctrl C                  to get back to BASIC
RUN                     to start up COPYA again
```

Follow the prompts and copy both disks.

2. Make the following sector edits to the copy.

Trk	Sct	Byte	From	To
$00	$02	$9E	D5	DE
$00	$03	$35	D5	DE
$00	$03	$91	D5	DE

Details

These programs were embarrassingly easy. There was a single technique used for copy protection and it was probably one of the oldest and simplest — changing the epilogues to the address and data portions of the sector. Perhaps it is yet another sign that companies are tiring of the "game" and are going to be more cooperative in the future, especially in the field of education.

Any one who is at all familiar with BASIC could have some real fun as most of the files are written in BASIC on these disks. Changing the way some of the files work and perform would certainly add a local flavor to the stories and routines. Enjoy!

Softkey for...

Kidwriter 1.0

Spinnaker

Requirements:
Fast copier
DOS (a fast one helps)
a way to change the boot program

Since none of the previous Spinnaker information helped me with this one, I thought I would send this in just in case someone else out there had the same problems I had.

Step-by-step

1. Fast copy the disk (ignore the error on track $00).
2. Copy all the files from your copy to another disk that has your DOS already on it
3. Use your boot program changer to change the boot file from "G" to "J". (See below.)

Details

The trick to getting this program working properly is being able to work around the control characters in the files. If you are using a copy program other than Copy II Plus, please modify the explanation accordingly.

The fast copy part of the deprotection is very straight forward. Ignore the error on track $00. Once you have your fast copy, tell Copy II Plus to CATALOG the disk showing the hidden characters. Surprise! How about that mess of sneaky glitches! Not to worry. When we copy these files to another disk using Copy II Plus, the file names are left intact and we don't have to worry about getting the proper control characters in the right places.

So initialize another disk with your favorite DOS (I noticed that they used Pronto DOS. You may have to try a different one if your favorite doesn't work. I used a patched DOS 3.3 and it seemed to work fine.) Then copy all the files from your fast copied disk to this new disk. As I said before, Copy II Plus handles the control characters in the file names just fine.

Once you are finished copying all the files over, tell Copy II Plus that you want to CHANGE THE BOOT PROGRAM. When it gives you the list of files, highlight the "J" file and press the "G" for GO. (The reason I placed this file in quotation marks is because the file really isn't just a "J" but has several control characters surrounding it. But it appears as a "J" on the file menu.) This procedure bypasses the regular boot file which appears on the menu as a "G" and also contains the signature check routine. Leaving it out simply puts the program into motion without the disk check and the program now boots up faster. You now have a perfectly COPYA-able deprotected KIDWRITER that you can safely allow children to use (without all those nightmares about replacement costs and delays!).

Softkey for...

Cloze Plus Program

Milliken

Requirements:
COPYA
sector editor

This program is a "context analysis" reading program made up of 6 sets of diskettes with 4 diskettes in each set. Each set of diskettes is designed for a specific reading level, starting at reading level 3 and proceeding up through reading level 8. Each diskette has a Manager routine built in such that the teacher can keep track of the progress of up to 100 students. It is an impressive program but has two flaws:

1. The program and the student data are all on the same side of the disk. If a student "diddles" the disk, the teacher loses all his/her data!

2. The program is copy protected which multiplies the hazard in 1.

I can't do too much about #1 except recommend you keep backups of the deprotected disk. And here is how to obtain your deprotected disk.

Step-by-step

1. Copy the disk using COPYA and ignore address and data epilogues.

```
POKE 47426,24
RUN COPYA
```

2. Make the following sector edits:

Trk	Sct	Byte	From	To
$11	$00	$01	F1	11
$00	$02	$9E	DF	DE
	$03	$35	DF	DE
	$03	$91	DF	DE
	$06	$AE	DF	DE

Details

The deprotection of this disk was very standard except for two details. The first detail is that the first sector edit above fixes the catalog such that files (especially those that contain important student data) can now be removed from the disk on a regular basis if necessary (or files can now be manipulated in any number of ways by enterprising teachers). Secondly, the teacher handbook that comes with the program draws special attention to the speed of disk drives and how important the proper speed is to the operation of the program. I was suspicious about this warning and determined that I would try

it out once I had the deprotection intact.

I found the disk speed check on track $00, sector $0A at the end of the sector. As the program checks for the standard Address and Data marks, it goes on to check for two FF bytes. I thought to myself that this would be a rather ingenious way of slipping in a protection scheme under the disguise of a disk speed check and consequently fooling the apprehensive hacker. I decided the best way to check would be to change my disk drive speed, edit the routine out of a copy, then see if the original disk and the copy gave the same results. I did just that. However, to my surprise the routine proved to be genuine. So all I will say now is if anyone wants to rid their program of the check and live dangerously or if you have a touchy disk drive that you are quite certain is not going to be a problem with this program, simply replace the D0 10 at bytes $F1-F2 and D0 07 at bytes $FA-FB (both on track $00 in sector $0A) with EA's. (Keep an eye on the re-boot instruction at byte $0E of sector $0B, track $00 as well!)

Softkey for...

Mighty Math

Discovery Software/World Book

Requirements:
COPYA or COPYA+
Sector editor (I use COPY II PLUS)
DOS System Master

Note: COPYA+ is an excellent tool from COMPUTIST #67 page 20, compliments of Gerald Berry and well worth the time to type it in.

Mighty Math is an excellent tool for teaching mathematical concepts and for drill exercises in basic math. It is designed for students at the primary level (ages 6-10) and has excellent options (varying speed levels, varying difficulty levels, and six different activities) for keeping the program interesting. It is one of the few programs that is worthwhile in the field of education. So many are a complete waste of time and money. The one problem with this program is that it is copy protected, so let's take care of that right now.

Step-by-step

1. Boot your DOS SYSTEM MASTER and RUN COPYA (or use your COPYA+ to achieve the same results as explained below)

```
RUN COPYA
ctrl C                  At the first prompt
70                      to prevent reloading COPY.OBJ
CALL-151                enter the monitor
B988:18 60              to ignore address epilogues
BE48:18                 to ignore errors
ctrl C                  to get back to BASIC
RUN                     to start up COPYA again
```

2. Follow the prompts and copy the disk.

Note: Track $03 has a special format that will not copy using COPYA. But since we have instructed our program to ignore errors, COPYA will "rattle" over the track - 32 "rattles" in all as it tries each sector twice. Do not be alarmed at the noise. Go ahead and count if it makes you feel better! Once it passes this track, all should proceed very quietly and efficiently.

3. When the copy is finished, start up your sector editor.

Trk	Sct	Byte	From	To
$11	$00	$01	00	11

4. Write the sector back to the disk.
5. Using your System Master or the Delete function on your COPY II PLUS, delete files HELLO, SSPROT$$A, and SSPROT$$1.
6. Use your System Master to get into BASIC. Replace your System Master with your new copy of the program. Type the following HELLO program:

```
10 HOME: POKE 34,23
20 PRINT CHR$(4)"MAXFILES1"
30 PRINT CHR$(4)"BLOAD SPEED1"
40 PRINT CHR$(4)"BLOAD SPEED2"
50 PRINT CHR$(4)"BRUN MAIN"
60 END
SAVE HELLO
```

7. Finally copy the DOS from your System Master to the new copy of the program that you have been making, or better yet, copy a fast DOS to the new disk.

Details:

This program gave me four hours of trouble. It was easy to get the COPYAed version. A quick look from the Nibble Editor of Copy II Plus showed the changes in track $03 and the epilogue changes to the address header on track $11. When I spotted the files SSPROT$$A and SSPROT$$1 it rang a bell in the back of my mind from a previous encounter I had several months ago with Walt Disney's Card and Party Shop. Since I didn't at that time have the back issues of COMPUTIST #50 and #51 to help me, it was a memorable wrestling match that took a full day to win.

This time I did some homework — some research with some of the best material available — back issues of COMPUTIST. Edward Teach in COMPUTIST #51 has some important observations with his encounter with the "SS" files in Walt Disney's Cartoon Maker. Also his HELLO program is useful. Secondly, A.L. Head's detailed explanation of the purpose and nature of the "SS" files in Walt Disney's Card and Party Shop is invaluable. Many thanks to both of these excellent operators for their help.

My own experience with "SS" files was limited to a BASIC version of both. I had not seen a binary form of either until I encountered this program. As I checked them out under the Sector Editor and disassembled them, it was obvious that they were not only in binary form but they were also encrypted! Hoping that they simply performed the same function, I took a guess and deleted them. Next I chose the first file from the list in the catalogue and (after loading a regular DOS) typed BRUN BOOT. The file loaded and I was then greeted with the final screen and music that ended the program! Bad guess!!

Next I simply did some more guessing. From what I had gleaned from Ed Teach, A.L. Head and others, it was simply a matter of trial and error until I got the correct combination. Nothing seemed to work, however. I even tried backing up and undeleting the "SS" files and then disabling their functions. Again, a bad choice!

Finally it occurred to me that the only two files on the whole list that were locked were SPEED1 and SPEED2. I did the BRUN on SPEED1 and was landed in the monitor. In an effort to circumvent reloading DOS and going through the same procedure with SPEED2, I simply used a **ctrl C** to get out of the monitor and back into BASIC, a CALL -25153 to connect the DOS that I hoped would still be intact, and then I typed the command BRUN SPEED2. This time I was met with a BASIC prompt. I was elated. All I had to do now was to guess the right file that would come next and I was home free. From some of my previous efforts I remembered seeing that the file called MAIN was one of the ones that the others often defaulted to at the end of their routines. So I typed BRUN MAIN and away the program went!

Next I quickly wrote a new HELLO file that summarized what I had learned and what Ed Teach and A.L. Head had also observed. I saved it to disk and then booted the disk. It sang like a riverboat sweetheart!

Softkey for...

Cotton's First Files

Mindplay

Requirements:
Fast copier (I use Copy II Plus)
Sector editor (I use Copy II Plus)

This program is a data base management system for beginning readers aged 4 to 9. Good start in the philosophy of data bases.

Step-by-step

1. Fast copy the disk (ignore the read error on track $0E).
2. Sector edit:

Trk	Sct	Byte	From	To
$01	$01	$02-03	A9 01	18 60

Or search for 60 A9 01 8D 27 B7. Don't forget to write the sector back to disk. Enjoy!

Softkey for...

French - Basic Vocabulary Builder on Computer

National Textbook Company

Requirements:
a way of capturing the disk's RWTS
Super IOB
fast DOS (preferable but not necessary)
slave disk (initialized DOS 3.3 disk with Hello file deleted)

This is an excellent program for teaching and reviewing written vocabulary skills in French. (This company also has programs in Spanish, German, and Italian.) The program uses graphics, help, and context clues to assist the student in learning the written language. It is a good diversion from the regular classroom routine.

Step-by-step

1. Boot the original and break into the monitor to capture the RWTS.

```
1900<B800.BFFFM    move RWTS to a safe place
```

2. Warm boot the slave disk.

```
6 ctrl P
```

3. Replace the slave disk with the disk that has the Super IOB program on it.

```
BSAVE RWTS.FRENCH, A$1900, L$800
RUN SUPER IOB
EXEC SWAP.CON
10010 PRINT CHR$(4) "BLOAD RWTS. FRENCH, A$1900"
RUN
```

4. Follow the prompts and copy the disk.

Details

There are two disks in the package - Disk 1 and Disk 2. You must capture the new RWTS from Disk 2 in order to copy Disk 2. But the procedure is identical and the copying is smooth and clean in both cases. I did have a problem with the opening graphic. In both cases this was scrambled for some reason. However, the rest of the program seemed unaffected. Both disks are almost impossible to back up with copy programs, and both disks are encrypted so as to make the contents completely unintelligible. Once this "garbage" has been cleared away, files can be examined and modified if desired. Happy French lessons!

Softkey for...

Life Coping Skills Series:
Forming Positive Behavior
Increasing Self Esteem
Building Relationships
Communication Skills

Plato/Control Data Corporation

Requirements:
initialized disks (as many as you have programs - initialize with a fast DOS for best results)
DEMUFFIN PLUS

These programs all belong to a set called the Life Coping Skills Series. Each title is a kit of anywhere from 2 to 7 disks with a manual that is helpful but not necessary. The disks all contain information and instruction with a few simple graphics. It is very informative but not very "inspiring"! Removing the protection was the same for all the disks in all the kits.

Step-by-step

1. Boot one of the disks in the kit and use your favorite method to reset into the monitor.

```
6000<B800.BFFFM    move the RWTS to a safe place
```

2. Remove the original disk and replace it with an initialized disk with HELLO deleted.

```
ctrl P             to boot the disk from the monitor
```

3. Remove the slave disk and replace it with your disk that has DeMuffin Plus on it.

```
BLOAD DEMUFFIN PLUS
CALL -151          to get into the monitor
B800<6000.67FFM    move the RWTS back where DeMuffin+ can use it
803G               to start up DeMuffin Plus
```

4. Choose CONVERT FILES, follow the prompts, choose the wildcard for the file name, and answer "N" to the "DO YOU WANT TO BE PROMPTED?" question.

The files will copy smoothly and easily and when the process is done, so is the deprotection!

Details:

The only detail that is important is that you must capture a new RWTS for each kit, but the same RWTS can be used for all the disks in each kit. Also I did notice that the kit called Building Relationships does not run on a Laser computer. After loading and listing the Applesoft files, I'm quite certain that by changing a couple of the CALLs in the program, all would run well on the Laser machines. Since I do not know enough about CALLs and POKEs to really offer much help, I'll leave this finer point to someone like Marc Batchelor who laughs at the simplicity of these tasks!

Softkey for...

Scrambled Words

Learning Well

Requirements:
COPYA
sector editor (I used Copy II Plus)

This program is designed to teach and reinforce alphabetizing skills. It offers an excellent alternative to the traditional drill. There are 4 skill levels and the word lists can be personalized. Another excellent Learning Well program!

Step-by-step

1. RUN COPYA and use your favorite method to ignore Address and Data Epilogues and Checksums.
2. Copy the disk.
3. Sector edit:

Trk	Sct	Byte	From	To
$00	$02	$9E	BF	DE
	$03	$35	BF	DE

That's it!

Softkey for...

Puzzle Tanks
Challenge Math
Explorer Metros: *A Metric Adventure*
The Whatsit Corporation: *Survival Math Skills*
Number Quest: *Binary Search Games*
The Kings Rule: *Mathematics & Discovery*
Incredible Laboratory: *Strategies in Problem Solving*
Fun House Maze: *Strategies in Problem Solving*
Code Quest: *Practice in Problem Solving*
The Pond: *Strategies in Problem Solving*
The Factory: *Strategies in Problem Solving*

Sunburst

Requirements:
original disk
initialized disks - HELLO deleted (use a fast DOS)
Super IOB with the Swap Controller
Copy II Plus (or similar utility for changing boot files)

All of the following Sunburst titles can be deprotected with the same RWTS captured from any one of the disks:

Step-by-step

1. Use your favorite method to capture the RWTS from any one of the above programs. Save it to the same disk as your Super IOB program.
2. Run your Super IOB and install the Swap Controller.
3. Change line 10010 to PRINT CHR$(4) "BLOAD RWTS.xxxx, A$1900" (fill in the xxxx with the name you gave the RWTS file you BSAVEd earlier).
4. RUN the program and follow the prompts.

You can use the same RWTS for all the disks. When one is finished, just be sure to place your Super IOB disk back in the drive

before you type the RUN for the next disk so it can load the RWTS again.

5. Finally use your Copy II Plus to change the boot program on all the disks to the first file in the list (for most of the programs the file is called LOGO). Occasionally there will be a disk that already has a HELLO file. These can be left as they are.

Softkey for...

Terrapin Logo Language

Terrapin, Inc.

Requirements:
original disk
Super IOB with Swap Controller
initialized disk

Step-by-step

1. Boot DOS (from your System Master or wherever) and INIT a slave disk with a binary hello name.

POKE 40514,52 *so DOS will BRUN the initialized file*
INIT SDOS
DELETE SDOS

2. Capture the RWTS of the Terrapin Language disk using your favorite method and BSAVE it to your Super IOB disk.
3. Remember to retype line 10010 to BLOAD the file you have just saved.
4. RUN Super IOB and EXEC SWAP.CON
5. Follow the prompts and copy the disk. (Answer "N" to the FORMAT THE BACKUP? question.)

That's it! Because of the nature of the program, adding a fast DOS doesn't make a great deal of difference. The Utilities Disk that comes with the Language Disk isn't copy-protected. The monstrous manual is excellent!

Softkey for...

Tic Tac Show

Advanced Ideas

Requirements:
COPYA (from the System Master disk)
a sector editor (I use Copy II Plus)

This is an excellent program that is akin to the television game show Hollywood Squares. One or two players can play and X's or O's are earned on the board by correctly answering questions about a selected subject area. Data disks with the questions already prepared may be purchased separately or (and here is the big plus for teachers) disks can be created on your own with your own subject areas and questions. This makes an excellent teaching and review tool for any kind of factual material. Now for the "fix" to make the program truly valuable in the classroom (or wherever).

Advanced Ideas uses three protection techniques on the disk - tracks $00 to $10 are all labelled track $00, secondly, address and data epilogues on all sectors from track $00 to track $10 have been changed from DE AA to FF FF, and thirdly, there is a signature check on track $09 to keep you on your toes.

Step-by-step

1. Boot your DOS 3.3 System Master disk and run COPYA.

RUN COPYA
70
ctrl C *at the first prompt*
POKE 47426,24 *to ignore epilogues*
POKE 48584,71 *to ignore the track checker*
RUN

2. Copy the whole disk (there should be no "grinds" or other problems)
3. Make the following edits:

Trk	Sct	Byte	From	To
$00	$04	$CE	FF	DE
$00	$05	$84	FF	DE
$00	$05	$8E	FF	AA
$09	$02	$07-09	BD 89 C0	60 60 60

Details

Perhaps the most important rule of disk cracking is to do a careful analysis of the disk prior to attempting any copying procedure. I wish I could remember that rule a little better. Disks like this one certainly underscore the importance of the procedural basics! I spent nearly two hours of "spinning my tires" with this program. After settling down and going back to Square 1, the whole thing was done 30 minutes later!

The Nibble Editor of Copy II Plus uncovered the altered epilogues. The Trax program in Bag of Tricks II highlighted the repetition of Track $00 (as well as the altered epilogues). After discovering these alterations, getting a copy using COPYA was easy. Then after making the sector edits from the FF FF back to DE AA in the appropriate places so DOS does not stumble while reading the disk, the loading "music" then gets an attentive ear. When you boot the disk at this point, you will hear the program begin to load, pause, then you get the scrambled screen and the reboot. Both of these conditions are symptomatic of a signature check.

So load up your sector editor and scan for the 8C C0 that is very likely some "distance" beyond DOS since we did hear some loading taking place after DOS was in place. Sure enough on track $09 the program checks for some E7's (C9 E7) after a disk read instruction. Further, the program returns (60) without any other special loading of the A or Y register. With that in mind, we may simply back up to where the program gives the instruction to turn on the disk drive to start this whole procedure (BD 89 C0) and tell it to return without bothering to do the check (60 60 60). And lo and behold, the program not only runs, but does so much more quickly since it does not have this whole routine to go through before actually starting the game!

Marc Batchelor _____ FL

Softkey for...

Where in Time is Carmen Sandiego (//e)

Broderbund

I just received Computist #74 and wow am I disappointed. I was hoping to have this crack in to Computist by issue 75. Even though the crack is already out (good job Brian and Terry), perhaps this may show that there is more than one way to skin a cat (or crack a ware).

Sorry for the lateness, but we're caught up now so anything new will be printed in the current or next issue. .....RDEXed

The Protection

Where in Time is Carmen Sandiego (Time) has a rather annoying protection. That is to say, finding the protection is fairly time consuming, removing the protection is simple, testing the product is a pain in the tush. The protection is not verified on startup, the disk is not perverted, the format is ProDOS and therefore hard drive compatible... it is... a key disk (oh no!). The only time that the protection shows up is when it is time to be promoted. At this time, the original disk needs to be inserted and verified by the software. To get promoted, you must solve a case (not exactly a piece of cake).

Locating the Protection

After playing the game for approximately 35 minutes, I finally caught the thief (whew!), and received the "Put disk 'A' in the drive" message. At this point, I pushed the switch on my Senior Prom, pressed '5' (All Main to Aux), and then looked at where it was waiting. It had stopped somewhere in the AEA0 page. Listing out the code here showed the following:

```
AEA?    LDA  $C000
        BPL  -03
        BIT  $C010
        RET
```

Essentially, this routine continuously monitors the key latch ($C000) until a key is pressed, it then clears the strobe ($C010) and returns. Searching memory for references to this routine (via Senior Prom), turned up a reference at $71BE, $A049 and $A09B. Listing at $71BE revealed:

```
71BE-   JSR  $4DE9
71C1-   JSR  $AEA0   <<- What we were just
                        looking at.
71C4-   JSR  $A8F4
71C7-   JMP  $719C
719C-   JSR  $9600   <<- The Protection
                        Routine.
719F-   BCC  +25
```

I discovered that typing 719CG turned on the disk drive, and then revealed the "Put Disk 'A' in the drive" message. Following further found the JSR to $9600. Which is the actual protection check.

Where is it at?

I searched side 'A' and side 'B' for the protection without success. I then started to break-out my 'EOR Scanner' and search for it when I decided to search the "data" disks. Low and behold, I found the JSR $9600 on side 'C', Track $1F, Sector $F, byte $9C and side 'D', Track $20, Sector $E, Byte $9C. Changing the JSR $9600 to $18 EA EA worked like a champ. Extensive testing revealed that the pause for an original disk never shows up. This version can and has been played right up to acceptance into the hall of fame.

Recap

Search side 'C' & 'D' for 8C 35 44 20 7D 7C 20 00 96 and change the 20 00 96 to 18 EA EA. That's it.

To Dave Grenda: (re: DOS 3.3 to ProDOS) Regarding your concerns about large DOS 3.3 programs receiving NO BUFFERS AVAILABLE message. This will occur on files that have a starting address under $800. A great deal of single-load games have been cracked such that the code has been captured from $800 and above, and at $7FD is a JMP to the "un-packing" routine that places all of the code in the correct places. Regardless of size, ProDOS has an built in safety feature that prevents BLOADing a file with a starting address below $800. This would overwrite/destroy certain screen "holes" where important variables are stored.

The fix

This is not for the faint-hearted. First, verify the load address of the game by cataloging it with COPY][+ "show file lengths" option. You should see a starting address of $7FD (usually). If this is the case, boot a standard DOS 3.3 disk and BLOAD the file. Take out a piece of paper and a pencil and be prepared to write. Enter the monitor:

CALL-151
7FD.802

Write down the series of bytes that you see. Separate the first three from the second three. The first three should be: 4C XX YY. This is the JuMP to the "un-packing" routine. The next step is less straight forward and requires some machine language experience. Disassemble the code with:

YYXXL

You are probably looking at the move routine that places everything in the correct places. If the code doesn't terminate with a JMP GGHH, type 'L' again. Keep doing this until you have reached the end of the routine (indicated by a JMP GGHH). The GGHH is important as well so write it down. Now, assuming that you have space at the end of the move routine to add the following code, you can type the following:

800:4C XX YY

Then, at the end of the move routine, place the following pseudo-code.

ADDR:A9 aa 8D 00 08 A9 bb 8D 01 08 A9 cc 8D 02 08 4C HH GG

ADDR = address that the 4C HH GG was located on; aa is byte 4 that you wrote down; bb is byte 5 that you wrote down and cc is byte 6 that you wrote down.

Assuming that you were able to follow the above instructions, this will solve your problems.

To Kathi Quan: (re: Didatech programs) In this issue (hopefully) I outline and provide the crack for Fay: Word Hunter and Fay that math woman. I hope these help.

Regarding your Pirates! problem, I experienced similar problems (and worse ones) using Copy II+ 9.0 on a IIgs. I wrote Central Point software and they sent me a free update. The update fixes some of these problems, but if I were you, I would use an earlier version until CPS gets some bugs worked out (like maybe a file compare routine that actually finds differences between files!?).

Regarding the 'D5 AA 96', 'DE AA', and 'D5 AA AD' questions, here is a VERY brief explanation of these bytes:

The 'D5 AA 96' are called Address Headers. These headers are the Disk Operating Systems (DOSs) way of knowing when a sector on a disk begins. The 'D5 AA AD' are called Data Headers. These headers are the DOSs way of knowing when the actual sector data starts. The 'DE AA' are called address/data epilogs. These tell DOS that this is either the end of the data or the end of the address field.

While this sounds cryptic, a fair analogy is that of a record. A record is composed of several songs. These songs are analogous to individual sectors on a disk. Preceding each song is a 'leader' or a 'pause'. This pause is analogous to the sector header. The sector header on a disk tells the operating system which track this is, and which sector it is — as well as a few other tid bits (tid bytes?). Anyway, to get to a particular song on a record, the user must lift the record player arm, and place it on the header of the next song.

The disk drive however reads the disk until it encounters the address headers (mentioned above). To the disk drive, this is the same thing as the leader on a record. The disk drive then makes sure that this is the correct track/sector (by reading the bytes after the address header), and if it is, reads the disk until it encounters the data headers. On a record, this is the 'end of the pause'. Once the data headers are read, the operating system 'knows' that it is reading data. But it needs some way to 'know' when the data ends (the next pause). This is the purpose of the data epilogs. When the operating system encounters these special markers, it 'knows' that the last piece of data has been read.

Mind you, the above is a condensed version of what actually happens, and it has its shortcomings. The authoritative source for the full picture (and I dare say a better explanation) is Don Worth and Pieter Lechners marvelous publication Beneath Apple ProDOS.

Back to the question presented however... Software publishers really enjoy playing around with these special markers. So much so that it is UNCOMMON to purchase a commercial piece of software that does NOT have alterations in these markers. The reason that changing these markers is so much 'fun' for them is because it is an easy first step in deterring the novice from prying into their code. Since normal DOS is looking for 'D5 AA 96', and a publisher changes their operating system to look for 'D4 AA 95', a normal operating system cannot properly read (without modification) their disk. I hope this has provided some insight into what some of the mumbo-jumbo means.

To Steven T. Romanoski: (re: Sticky Bear Series) If it some consolation to you, I have cracked what I believe to be the entire series. If you are interested, send me your originals for about 2 days, and I'll send them back with cracked copies. Please send a blank disk with each. My address was published in Computist 73.

Sierra On-Line generically used/uses two protections. One is a nibble count/signature check, and the other is a code checksum routine that detects code alterations (if not performed correctly).

Softkey for...

MuppetVille
The Muppet Word Book
Memory Building Blocks

Sunburst Communications

The above three packages by Sunburst Software can be de-protected using ProDOS IOB, and a slight modification to the controller.

1. Boot the software package. When the boot is complete, break into the monitor, and type in the memory move given in the ProDOS IOB article.
2. Boot your ProDOS IOB disk, move the controller down in memory and save it (outlined in the article).
3. Load in PRDS.IOB, and EXEC SWAP.PCON.
4. Edit line 2010 and add: ':GOSUB 280' (ignores un-readable blocks).
5. Edit line 10010 to read the correct name of the driver (saved in step 2).
6. Run.
7. When complete, copy a new ProDOS onto the disks.

That's All there is to it.

An closer look at the protection on Fay's Word Rally

Preliminary Examination

When I received this disk to crack from a good friend, my first impression was that it would be a cinch. Upon booting, it had a DOS prompt, it sounded normal, and reset could be pressed without the program re-booting. Using a fast copier yielded an error on track 2, sector 7, and the rest of the disk copied normally. Booting the copy resulted in a cleared screen and a DOS prompt after displaying the "Loading" message.

In Depth Examination

Since I religiously use Pronto DOS for as many programs as possible, my next step was to format a disk under Pronto DOS using POKE 40514,52 (load binary "hello" program") and INIT DV (DV is the binary "hello" program). I then copied all files from the original to the formatted disk (don't forget side 2). Booting this copy landed me into the monitor at $BCE1. Since this is normally an un-allocated region (per Beneath Apple DOS), I examined this area on the original disk (track 0, sector 6, byte E1). What was stored here was an EOR routine used to decrypt the data stored at $6000. On track 0, sector 4 was a short routine to remove this code which is called after decryption.

The binary file FWR is loaded by DV to $6000. Examining this file did show that it was encrypted, so I decrypted the file (using their routine) and re-saved it. Then, I needed to patch DV to jump into FWR instead of jumping to the decryption. After this was done, I booted my new copy. This had the same result as my fast copied version (cleared screen and a DOS prompt).

My next step was to search the file FWR for BD 8C C0. No occurrences were located. The next step then was to trace the code (uggh!). Listing from $7893 (the entry point to the code), I traced until I found this beauty:

```
85E8:A0 00    LDY  #$00    Set up indirect index to
                           $5000
85EA:84 56    STY  $56
85EC:84 1B    STY  $1B
85EE:A9 50    LDA  #$50
85F0:85 57    STA  $57     56.57 = $5000
85F2:A9 BD    LDA  #$BD    First part of BD 8C C0
                           (Read disk).
85F4:20 B5 86 JSR  $86B5   Store throughout $5000
                           page via 56.57
85F7:A9 8C    LDA  #$8C    Second part of BD 8C
                           C0 (Read disk).
85F9:20 B5 86 JSR  $86B5   Store throughout $5000
                           page via 56.57
85FC:A9 C0    LDA  #$C0    Last part of BD 8C C0
                           (Read disk).
85FE:20 B5 86 JSR  $86B5   Store throughout $5000
                           page via 56.57
8601:A9 8D    LDA  #$8D    First part of 8D C0 50
                           (Storage).
8603:20 B5 86 JSR  $86B5   Store throughout $5000
                           page via 56.57
8606:A9 C0    LDA  #$C0    Second part of 8D C0
                           50 (Storage).
8608:18       CLC
8609:65 1B    ADC  $1B
860B:20 B5 86 JSR  $86B5
860E:A9 50    LDA  #$50    Final part of 8D C0 50
                           (Storage).
8610:20 B5 86 JSR  $86B5
8613:E6 1B    INC  $1B
8615:A5 1B    LDA  $1B
8617:C9 1E    CMP  #$1E
8619:90 D7    BCC  $85F2
861B:A9 60    LDA  #$60    Store $60 (RTS) at end
                           of check)
861D:20 B5 86 JSR  $86B5
8620:20 E3 03 JSR  $03E3   Find IOB
8623:84 CE    STY  $CE     Store Location in CE.CF
8625:85 CF    STA  $CF
8627:A9 02    LDA  #$02    Track 2
8629:A0 04    LDY  #$04    Position in IOB for Track
                           Number
862B:91 CE    STA  ($CE),Y Use CE.CF indirectly to
                           rebuild IOB
862D:A9 00    LDA  #$00    Command = SEEK
862F:A0 0C    LDY  #$0C    Position in IOB for
                           COMMAND CODE
8631:91 CE    STA  ($CE),Y
8633:A0 03    LDY  #$03    Position in IOB for
                           Volume ($00 = any)
8635:91 CE    STA  ($CE),Y
8637:20 E3 03 JSR  $03E3
863A:20 D9 03 JSR  $03D9   Seek Drive to Track 2
863D:B0 27    BCS  $8666
863F:BD 89 C0 LDA  $C089,X Turn On Drive
8642:A9 30    LDA  #$30    Only do this $29 Times
8644:8D 78 05 STA  $0578
8647:38       SEC
8648:CE 78 05 DEC  $0578
864B:F0 19    BEQ  $8666
864D:20 44 B9 JSR  $B944   Read Address Field
8650:B0 F5    BCS  $8647
8652:A5 2D    LDA  $2D     Sector Number
8654:C9 01    CMP  #$01    Physical sector 1 (DOS
                           Sector 7)?
8656:D0 EF    BNE  $8647   Nope... Try again!
8658:BD 8E C0 LDA  $C08E,X
865B:A9 06    LDA  #$06
865D:20 A8 FC JSR  $FCA8
8660:20 00 50 JSR  $5000   Go run funky Signature
                           Check
8663:18       CLC
8664:90 04    BCC  $866A   Always Taken
8666:A0 0D    LDY  #$0D
8668:B1 CE    LDA  ($CE),Y
866A:9D 88 C0 STA  $C088,X Shut Off Drive
866D:A0 00    LDY  #$00
866F:84 48    STY  $48
8671:B0 38    BCS  $86AB
8673:84 56    STY  $56
8675:A2 00    LDX  #$00
```

At $5000 After 86E8-8620 is run.

```
5000:BD 8C C0 LDA  $C08C,X
5003:8D C0 50 STA  $50C0
5006:BD 8C C0 LDA  $C08C,X
5009:8D C1 50 STA  $50C1
500C:BD 8C C0 LDA  $C08C,X
500F:8D C2 50 STA  $50C2
5012:BD 8C C0 LDA  $C08C,X
5015:8D C3 50 STA  $50C3
5018:BD 8C C0 LDA  $C08C,X
501B:8D C4 50 STA  $50C4
501E:BD 8C C0 LDA  $C08C,X
5021:8D C5 50 STA  $50C5
5024:BD 8C C0 LDA  $C08C,X
5027:8D C6 50 STA  $50C6
...
50A8:BD 8C C0 LDA  $C08C,X
50AB:8D DC 50 STA  $50DC
50AE:BD 8C C0 LDA  $C08C,X
50B1:8D DD 50 STA  $50DD
50B4:60       RTS
```

This software actually builds the signature check on the fly. Notice however, that the signature check doesn't wait to make sure that it has read a valid disk byte — that is, there is no BPL following the read of the disk. This indicates a timing dependency to get the correct bytes read.

The Fix

To see the above code is easy... just boot the disk, and when the drive stops, press reset. Kidsoft did not think to clear the signature check (or the bytes following it) in the reset handler that they have. When reset is pressed, you will be placed in DOS (via $E000). At this point, we need to save the results of the check into a file (called NI). We just insert the copy, and BSAVE NI,A$5000,L$E0. The only thing left is to patch the FWR program to BLOAD NI instead of performing the nibble count, and branch past the call to the nibble count.

Softkey for...

Fay: That Math Woman

Didatech Software

Requirements:

Fast DOS which de-allocates most of track 2

Fay: That Math Woman Original Disk

File Copier

Step-by-step

1. Initialize a blank disk with Pronto DOS (or other DOS that de-allocates most of Track 2).

```
POKE 40514,52
INIT COPYRIGHT 1986
DELETE COPYRIGHT 1986
```

2. Use a file copier to copy the files (There is only one side).
3. Boot a DOS disk.

```
BLOAD COPYRIGHT 1986
CALL-151
846:4C 05 67          remove decryption call
AA60.AA61     -       file length in YY XX format
UNLOCK COPYRIGHT 1986
BSAVE COPYRIGHT 1986,A$802,L$XXYY
LOCK COPYRIGHT 1986
```

3. Load the encrypted FMWS program.

```
BLOAD FMWS
```

4. Type the following to decrypt the FMWS file:

```
2000:A2 35 A0 03 A9 62 59 00 60 99 00 60
200C:C8 D0 F5 EE 08 20 EE 0B 20 CA D0 EC 00
2000G
```

5. Determine the length of the FMWS file and save it.

```
AA60.AA61             length in YY XX format
UNLOCK FMWS
BSAVE FMWS,A$6000,L$XXYY
```

6. Place the original in drive 1, and type in the following to capture the signature check.

```
889E:00 N 6705G
```

7. Insert your copy and BSAVE the signature check to side 1.

```
BSAVE NI,A$5000,L$E0
14.
```

8. Patch FMWS to make it load the signature results. Save the patched code.

```
BLOAD FMWS
8823:A2 00 BD 30 88 F0 74 20 ED FD E8 D0 F5
8830:84 C2 CC CF C1 C4 A0 CE C9 8D 00
AA60.AA61             length in YY XX format
BSAVE FMWS,A$6000,L$XXYY
LOCK FMWS
```

That's All for Fay: That Math Woman

Softkey for...

Fay's Word Rally

Didatech Software

Requirements:

Fast DOS which de-allocates most of track 2

Fay's Word Rally Original Disk

File Copier

Step-by-step

1. Initialize both sides of a blank disk with Pronto DOS (or other DOS that de-allocates the part of track 2 that is not used by DOS).

```
POKE 40514,52         for binary hello
INIT DV
DELETE DV
```

2. Use a file copier to copy the files. (Don't forget to file copy side 2).
3. Boot the FWR disk.
4. When the drive stops, press RESET.
5. Insert your copy and BSAVE the signature check to side 1.

```
BSAVE NI,A$5000,L$E0
```

6. Boot a DOS disk.
7. Insert your copy, side 1.

```
BLOAD DV
```

8. Patch to remove decryption call to DOS.

```
CALL -151
934:4C 93 78
```

9. Determine file length.

```
AA60.AA61
```

10. You will receive the length in YY XX format.

```
UNLOCK DV
BSAVE DV,A$802,L$XXYY
LOCK DV
```

11. Load the encrypted FWR.

```
BLOAD FWR
```

12. Type the following to decrypt the FWR file:

```
2000:A2 35 A0 03 A9 53 59 00 60 99 00 60
200C:C8 D0 F5 EE 08 20 EE 0B 20 CA D0 EC 00
2000G
```

This disassembles to:

```
2000:A2 35    LDX  #$35
2002:A0 03    LDY  #$03
2004:A9 53    LDA  #$53
2006:59 00 60 EOR  $6000,Y
2009:99 00 60 STA  $6000,Y
200C:C8       INY
200D:D0 F5    BNE  $2004
200F:EE 08 20 INC  $2008
2012:EE 0B 20 INC  $200B
2015:CA       DEX
2016:D0 EC    BNE  $2004
2018:00       BRK
```

13. Make the following patches to FWR to load the signature results.

```
85E8:A2 00 BD F8 85 F0 74 20 ED FD E8 D0 F5
85F5:00 00 84 C2 CC CF C1 C4 A0 CE C9 8D 00
8660:EA EA EA
```

This disassembles to:

```
85E8:A2 00    LDX  #$00
85EA:BD F8 85 LDA  $85F8,X
85ED:F0 74    BEQ  $8663
85EF:20 ED FD JSR  $FDED
85F2:E8       INX
85F3:D0 F5    BNE  $85EA
85F5:00       BRK
85F6:00       BRK
85F7:00       BRK
85F8:"^DBLOAD NI^M"
```

14. Determine Length of FWR file.

```
AA60.AA61
```

15. You will receive the length in YY XX format.

```
UNLOCK FWR
BSAVE FWR,A$6000,L$XXYY
LOCK FWR
```

That is all for Fay's Word Rally.

Alan Chaney — MD

Softkey for...

Star RankBoxing II

Gamestar

Requirements:

1 side of a blank disk

copy program that ignores errors

sector editor

I copied the original program using copy 2+ fast copy. Upon booting the copied program I was greeted with (PLEASE PUT ORIGINAL DISK IN DRIVE). I then searched the disk for the PLEASE part of the message, and found it in 2 places both on track one in sectors 5 and 8. I noticed that there was a D0 01 that branched pass a RETURN (60) and seem to run the PLEASE part of the message. So I tried Noping out the above branch and you guessed it.

1. Copy the disk.
2. Search for A5 FF D0 01 60 2C and change D0 01 to EA EA. You should find it twice.
3. Write the changes back to disk. (Copy only.)

Softkey for...

Ikari Warriors II (Victory Road)

Data East

Requirements:

copy program that ignores errors

sector edit

I search my issues for a softkey to this program and found 2, one by Mr. Moolenaar issue #52, and one by Mr. Simon in issue #53, but both were for Ikari Warriors and not part II of the program.

So I copied the program (both sides) and booted the copy. Almost instantly the program rebooted again and again even before a program screen showed up. So out came issue #68 and Mr. Karwoski's crack bible for references. After reading page 29 a minute or two, I tried looking on the disk for the reboot code of 4C 00 C6. I found the code on

the disk and Noped it out. Altogether now, I can't be this smart or can I?

1. Copy both sides of the disk.
2. Search for F0 03 4C 00 C6 A9 60 and change the 4C 00 C6 to EA EA EA.

Put Carmen (USA) on a 3.5" disk

Requirements:
The deprotected 5.25" program.
one inited 3.5" disk
A program to sector edit with.

1. Name the 3.5" disk anything but Carmen.USA.
2. Create 2 subdirectories, one to be for side B and one to be for side C (both directories should each come off of your Volume name. For example:

```
/ANYTHING
 ANYTHING ->->-> ACARMEN.USA.S2
          ->->-> ACARMEN.USA.S3
```

3. Copy all files except ProDOS from side A to your Volume name set in 1.
4. Copy all files from side B to sub ACARMEN.USA.S2
5. Copy all files from side C to sub ACARMEN.USA.S3
6. Copy a clean ProDOS to the Volume section of disk.
7. Search disk for "/CARMEN.USA." (last period is a must).
8. We are going to make the Volume look for "ACARMEN" instead of "/CARMEN".
9. Change 2F or "/" to 41 or "A" (4 Changes only), (NOT block 29).
10. Search disk for lower case "/carmen.usa.". Same changes (2 changes only).
11. There should have been 4 "/C" and 2 "/c" changes for a total of 6 changes.
12. Rename Volume to "CARMEN.USA".

Time for you to go to work! Note: Hope this wasn't that hard to follow, because when Mr. Egnotovich of issue #72 performed the same thing it was simple enough for even the Wildman. Any questions? Call a CAB! Just joking.

Softkey for...
Where in Time is Carmen Sandiego? (GS)
Broderbund

Requirements:
One 3.5" disk
Fast copier
sector editor
Issue #72, Page 30 (by Mr. Hodge)

The sector patch for Where in the USA (GS) also works for Where in time (GS). Just scan for the bytes to be changed. The article appeared in issue #72 (pg 30) and it was sent in by Mr. Hodge.

On my disk the edit 8D to 60 is found easier by looking for A0 C0 8D 24 9A on block 31F at byte 189. The big sector edit was on block 320 starting at byte 1F.

Note: In the big sector edit EB was E8 on my disk.

I hope I don't get sued for Plagiarism, because my ship is stuck in the Persian gulf until further notice.

Here is the patch by Mr. Hodge of issue #72

Blk	Byte	From	To
$D7	$1DE	A2 32 A0 00	A2 48 A0 1E
		AD EE C0 10	A9 9F 8F DF
		FB C9 D5 F0	43 00 A9 20
		06 EB D0 F4	8F E1 43 00
		C8 D0 F1 AD	A9 76 48 28
		EE C0 10 FB	A9 00 18 60
	$148	8D	60

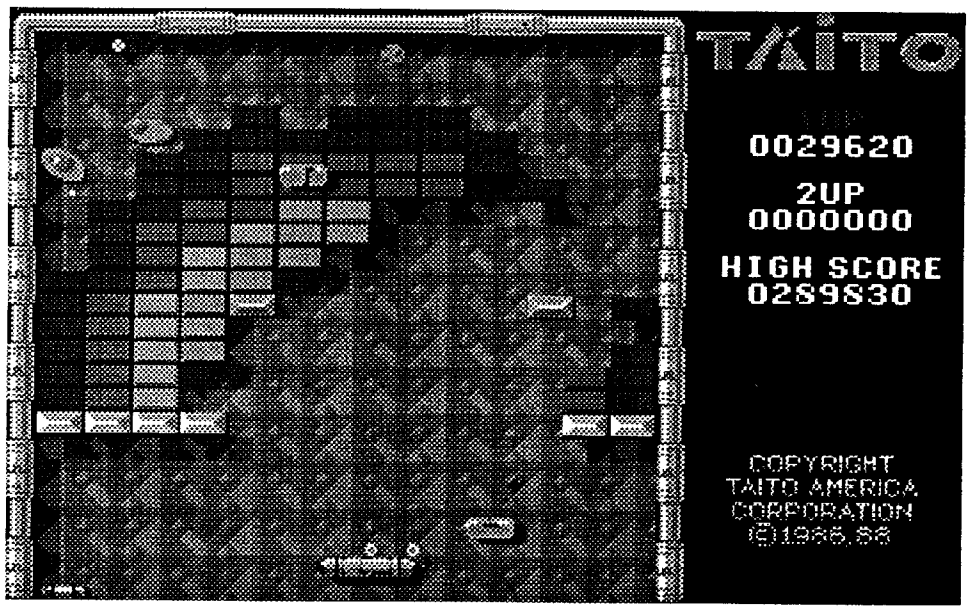
Put Carmen (Time) on a 3.5" disk

Requirements:
One 3.5" disk
Sector Editor
Original program disks

The Times disk has a file that will automatically copy the program to a Hard Drive, Ram disk, or 3.5" disk, the file if I can still remember is "IN". But the copy continues to ask for the original program every time you advance to other floors (which is a promotion in rank). So I decided to make the program truly hard drive compatible even though I don't own such a luxury.

The protection is in file MP. Scan the disk for 7D 7C 20 00 96 90 25 and change 20 to AD now the copy will promote you to the next floor without the original. I think I got it?

Note: The byte change also works on the 5.25" copies. Scan disk "C" for the string. (Should I smile? Not yet!).



Advanced Playing Technique for...
Arkanoid II: Revenge of Doh GS
Taito

Requirements:
Deprotected 3.5" copy of program.

Unlimited men

03/0425:DE E3 0F to EA EA EA. .

Make these changes in the Monitor.

03/0FE3:03	# of Men Player 1.
03/0FE5:03	# of Men Player 2.
03/1227	Highest score shown on screen.
03/A754	Where 5 high scores are kept.
00/1B0A:01	Level for present player.
00/1B28:01	Level for Player 1.
00/1B2A:01	Level for Player 2.
00/1B76-1B79	present score player 1.
00/1B7A-1B7D	present score player 2.

My subscription began at issue #69. You might think that there need to be more in the area of explanations, But this is all I have at this moment. Maybe after I get a few more issues under my skull cap, the articles will become a little easier to understand. Thanks goes to Computist and the many others that helped me to this point.

Softkey for...
Studymate (The grade booster)
Compu-Teach

Requirements:
Two 3.5" disk
3.5" Fast copier
3.5" Sector Editor

Copy both disks. The protection is on disk 1. The protection is a key word found in the manual. The program gives you the page, paragraph and the number of words to count to on that line. After typing in the word and pressing return the program starts. I tried with no luck to eliminate the check altogether. But I was able to make the program look in the same place every time for the key word to start the program.

Scan the disk for 20 22 FC 20 1B FD and change to EA EA EA EA EA EA. Now the program will always ask for 36, 1, 4 which is METHODS.

Help with Stickybears Disk (Optimum Resources)

Requirements:
Copy program that ignores errors.
Sector Editor
(maybe issue # 68 if you have problems)

This help is being brought to you in part by the fine article written by Ralph Supinski in Computist #68. I think that the particular area on a disk where a protection is located is important to some, but not all of us. This help will be in the form of scans for bytes, instead of track and sector locations. If you can't find these bytes on your disk, you will have to consult a back issue for your particular disk's softkey.

Step-by-step

1. Copy disk with any program that ignores errors.
2. Scan disk for 60 A2 00 A0 00 and change A2 to 60.
3. Scan disk for the following 8 bytes that apply to the disk that you may have. These bytes are found at the beginning of the sector in most cases ,but not in all cases. Once you find the string, copy that whole sector to sector $0F on track $01.

Math 1
CA D0 F3 88 D0 F0 18 60

Math 2
10 07 18 A0 05 71 75 A0

Parts of Speech
91 75 AA A0 01 B1 75 10

Reading
9B 1D 5F 32 56 A0 AD 56

Word Problems
A9 00 65 97 48 8A 48 60

Typing
53 68 29 F0 4A 4A 4A 4A

Town Builder
C8 84 4D 4C 0E 09 85 5F

Math Word Problems
85 97 A0 01 B1 96 F0 06

Spellgrabber
5E 29 20 F0 04 8D 55 C0

Reading Comprehension
00 00 00 00 00 00 00 00

Volcabulary Development
07 07 07 07 07 07 07 07

Spelling Rules
07 07 07 07 07 07 07 07

Sentence Fun 3.5
My copy not protected

Spelling Rules 3.5
B0 03 4C 00 00 *Change B0 to 80.*

Hope these quick keys helped someone? (I still don't have it yet.)

Softkey for...
Qix
Taito

Requirements:
One blank 3.5" disk
Fast copy program
Sector Editor

The Softkey I am about to give is actually Mr. Brian A. Troha's softkey of issue #73 pages 14 and 15. All I've done was to add 02 in the place of his 08. His article even told me how to find the 02 and where to put the number in the softkey. The man's a wizard. For further details see his masterpiece.

1. Copy original disk and make these edits to the copy.

Blk	Byte	From	To
4D5	77	22 73 02 00	AF 73 02 00
	81	22 A3 02 00	AF A3 02 00
	AC	22 B6 02 00	AF B6 02 00
	B0	90 09	80 09
5B4	DC	0C 07 03 09	04 04 04 00

2. Write changes onto your disk copy.

I am glad to be a part of this Organization. Am I a part yet?

Note on The Legend of Blacksilver (Epyx)

As of this writing, I have yet to find a protection on this disk. But I was kind of like hoping that a protection was there just to see if the first break was the start of some thing big?. I guess I have to buy another piece of software to work on. Don't get the wrong idea, I did play this game for an hour or so, and during that time the disk worked perfectly fine.

Advanced Playing Technique for...
Qix GS
Taito

Unlimited Men

Search your deprotected copy for DE 33 09 and change to EA EA EA.

Locations and changes

These are the edits you must make from the Monitor. Some could be on the disk, but some I know are not. All of these changes are in bank $04.

addr	change to/reason or function
1A0C:03	to 04, eliminates Sparx's from screen.
0949:01	to #, multiplies score that # of times for Player 1 (max 6E) than return to 01.
099B:01	to #, Player 2's timer.
093D-0940	Player 1's totals so far.
441C-441F	Player 1's totals so far.
0941-0944	Player 2's totals so far.
4420-4423	Player 2's totals so far.
3E26-3E29	Score of player now playing.
092B:01	Location of threst % Player 1
092D:01	Location of threst % player 2

Threst % is controlled by the # in the above locations. The numbers are 1-4=65%, 5-6=70%, 7=75%, 8-FF=80%. To keep threst at a constant % change 3A75:FE 2B 09 to EA EA EA.

092F:01	Levels Player 1.
0931:01	Levels Player 2.
0933:03	Holds # of men for Player 1.
0935:03	Holds # of men for player 2.
4B13	Location where 5 high scores are kept for score board. (# are read backwards).

Instant points

These 2 changes will allow any size block made on the bottom or base line to score 99% instantly. (Be careful not to get caught on the broken line part of block or it could get real ugly. (These addresses are in bank $03.)

80FC:80 0C	to EA EA
8103:80 0A	to EA EA

Steve Kalynuik _____ Canada

Playing Tip for...
Wizardry V: Heart of the Maelstrom
Sir Tech

Level 1

Ye castle teleport (12E,4N) Use bag of tokens.

Bag of tokens (4E,4N) Fight for it.

Temple of Brotherhood (8E,17N) Need to carry Orb of Llylgamyn.

Orb of Llylgamyn (18E,9N)

Silver Door (5E,27N) Use Silver Key to open.

Silver Key (27E,14N) Inspect hidden items.

Brass Key, Buy from Iron Nose in area behind riddle secret door.

Riddle (20E,27N) Answer Vampire.

Laughing kettle (22E,30N) Pay 100, 500, 1000, 2500, or 5000 for info.

Conveyor control (6E,4N) Press 'D B C A' to shut off.

Stairs to area of Den of Thieves (15E,9N).

When shut conveyor off, cast Litofeit, secret door (12E,1N).

Secret door (20E,6S).

Level 2

Ruby Warlock (7W,3N) Give bottle of rum to pass.

Bottle of rum (3W,14S) Search crate.

Jewel encrusted chest (4E,0N) Use spirit-away potion to get Jeweled Scepter.

Door sealed with heavy chains (2E,15S) Use hacksaw.

Hacksaw (4E,3N) Inspect hidden items.

Randolf's Mixture Emporium (8E,0N) Mix 'C A F' Get spirit-away potion.

Le Dragon Flagon Bar (13E,9N).

Bubba's Health Spa (9E,10N).

Duck of Sparks near Bubba's Health Spa, Give rubber duck, get munke wand.

Elevator to levels 2,3,4, and 5 (7E,4S).

Level 3

Quiet poll of liquid gold (26E,14S) Swim with no gold to level G, You will get gold and the gold key.

Wall of deepest blue (12E,25S) Use blue candle.

Grotesque figure overlooking fountain (2W,14S) Swim to level H, get petrified demon.

Temple of Kama Kazi (12E,5N) Fight Lord Heinmitey, then use Jeweled Staff to open door.

Timeless room (17E,8N) Inspect hidden items, use battery, press 'D E G C' get pocketwatch.

Blue Candle (12E,8N).

Mad stomper on level 3, buy from him the rubber duck.

Level 4

Foggy pool with sexy lady (9E,9S) Swim to level J, get skeleton key.

Old trunk marked D.H. (11E,7S) Inspect hidden items, get battery.

Nymphette sitting on chest (11E,16S) Fight, get jack of spades.

Face of demon on wall (6E,17S) Use petrified demon.

Den of Thieves (10E,0N).

Totem of many faces (1W,24S) Answer: time.

Massive door with ivory skull (4E,24S) Use skeleton key.

R.I.P. the loon (2E,24S) Use pocketwatch.

The loon buy bird in cage.

Ye gold vault (15W,21S) Use gold key.

Black circle on floor touch the following: 11W,33S / 9W,31S / 17W,34S / 17W,28S / 12W,28S / 8W,25S / 16W,30S

Black circle on floor do not touch the following: 9W,34S / 13W,28S / 12W,25S

Box with two statues (8W,21S) Search, fight, enter portal to hall of mirrors.

Level 5

Big Max (7E,1S) Buy tickets, give tickets or ticket stubs.

Playhouse theatre: Warning do not enter as no spells work here.

Secret door (6E,8S).

Secret door (9E,6N).

Secret door (2E,4S).

Pool of blue water (20W,1S) Swim or lower a character to level C and will bring to life, cure, and heal even a lost character.

Secret door (1W,15S).

Secret door (5E,15S).

The snatch (5E,12S) Give gold, learn about loon's locket.

Level 6

Dark well (8W,9S) Swim to level N, get queen of hearts.

Evil eyes near (11W,2S) Try to steal gold medallion.

Ice key (0E,2S) Inspect hidden items.

Strange contraption (Ferry to Ice Castle) (7E,18S) Press 'G D A F E B C' to fix.

Secret door (7E,17S).

Ice Castle temple (8E,25S) Must use ice ferry to get here, get king of diamonds.

Strange hole (5E,25S) Use ice key.

Beware of quicksand at (14E,11S) and (12E,9S) Cast Litofeit.

Frosted glass cylinder (14E,5S) Use gold medallion.

Level 7

Deep pool of midnight blue (10E,12N) Swim to level P, get staff of water, level o will bring the dead to life.

Large sparks arc (13E,4S) Need lightning rod.

Staff of fire (13E,9S).

Brass monkeys (0E,13S) Need munke wand.

Lord of Clubs - White Knight (5E,6S) Ask time, nature, kingdom. Give ace of clubs to pass.

Lord of Hearts - Blue Knight (5E,5N) Ask time, nature, kingdom. Give queen of hearts to pass.

Lord of Spades - Red Knight (6W,5N) Ask time, nature, kingdom. Give jack of spades to pass.

Lord of Diamonds - Yellow Knight (6W,6S) Ask time, nature, kingdom. Give king of diamonds to pass.

Howling wind (13W,0N) Need lark in cage to pass.

Speckled bird (12W,7S) Answer: life, get staff of air.

Staff of Earth (5W,13S).

In the middle of level 7 are four flames, for the color of knight you gave the card to, use Orb of Llylgamyn.

Level 8

Fight clones of self at (6W,0N) (0E,6N) (6E,0N) (0E,6S).

At (4W,0N) use Staff of Earth, press 'A D I'. Answer is from one of the Lords of Cards.

At (4E,0N) use Staff of Fire, press 'C F G'. Answer is from one of the Lords of Cards.

At (0E,4S) use Staff of Water, press 'B E H'. Answer is from one of the Lords of Cards.

At (0E,4N) use Staff of Air.

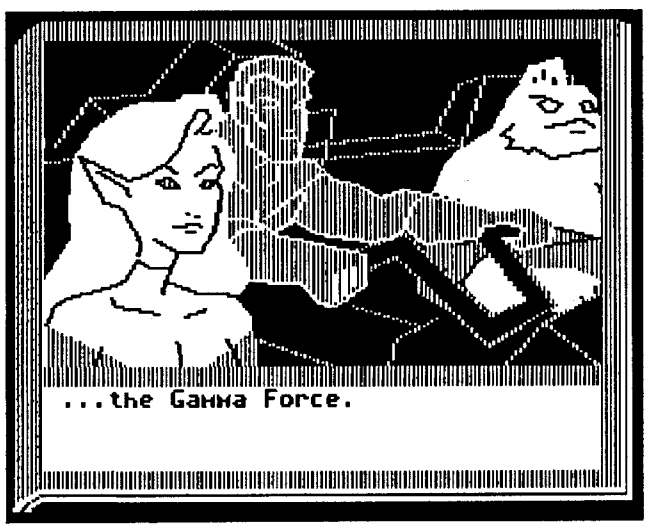
The rest is up to you, for you yourself have to understand, in order to become the 'Guardian's of this mortal plane'.

May the love of Abriel go with you.

One final note: Be wary of the 'Netherdemon' and 'Archdevil' in the level known as Hades.

Please write if you have any more questions.

Ⓟ Please would somebody contact me if they have or know about copies of 'Wizards Workbench' 'Wizards Workbench II' or 'Wizi-Scout' by Magicsoft. These are Wizardry scenario aides, and scenario creators.



Roland Boucher _____ CA

Softkey for...

Gamma Force
Zork Quest
Infocom

The copy protection routine in both these programs resides in Track $03, Sector $0E and I suspect that it is used in the other Infocomics. Search for the sequence: BC 8C C0 10 FB C0 FF and change the BC to 60.

Step-by-step

1. Copy both sides of the disk with any normal copy program.
2. Make the following sector edits to the boot side of the copy you have made:

Zork Quest

Trk	Sct	Byte	From	To
$03	$0E	$28	BC	60

Gamma Force

Trk	Sct	Byte	From	To
$03	$0E	$26	BC	60

3. Write the change back to the disk.

E.N. Hondrick _____ CA

To Bob Igo: (issue #73, p19) — Copy II+ does block read/write on both 3.5" and 5.25" discs, of course, and so does ProSEL's BLOCK WARDEN and several others. There's a stand-alone ProDOS block editor in issue #55, p14 or you can even decode blocks to Track/Sector numbers, using a table in the same issue as your letter, p11. Then too, doesn't Super IOB run under ProDOS now? You probably know all this already, so what am I missing? — if it's the ability to read sectors with queer headers, then I hope you find out how and tell the rest of us, because I can't see how it's done either.

Your question (issue #65, p18) about keyboard bounce may be discussed in Jim Sather's "Understanding Your IIe", but it's out of print and I can't get a copy—sorry. Try spraying around the key with WD-40.

To Groucho Tarz (issue #73, p19)—the modifications in Computist #71, p14 and #66, p21 sound good. Also change $F949 to a 1 or 2 to make a narrower listing and insert a JSR around $F8D0 to a patch that shows the ASCII equivalents during disassembly. Include a NMI or redirect the RESET vector from $3F2-3F4, quit "blotting out" two bytes per sector, and have the auto-repeat function use a shorter "wait" value or one stored somewhere in RAM; pick a byte that's loaded with ProDOS. Fix RESET so it doesn't kill all your variables! You might want to keep the TAPE routines so you can use "AppleEar" or a Cauzin Strip, but I would get rid of those useless mouse icons and restore inverse/flash characters or use the space to provide a XFER.BOOT or some memory-save function for pages 0-7. You may want to fix INPUT to accept commas and colons and make other changes, but think twice; most of these improvements will prove useless unless you're writing programs for exclusive use on the one machine.

To Randy Flood (issue #73, p18) **and J.P. Mulder** (issue #72, p10)—get the highest baud-rate modem available; you can set it to any lower rate you wish. Make it an external modem (you never know when you might decide to buy a new computer!) and get a communications program you're comfortable with that handles standard AT format: ProTerm is as good as any and better than most.

To Randy, Zorro (issue #72, p17) **and Gary M. Thorpe** (issue #67, p12)—the method described by Groucho Tarz (issue #73, p19) has been mentioned before in a very early Computist and elsewhere. If you only want a NMI card, your favorite magazine had some Integer cards for sale or I can probably buy you one here for the same terms plus a dollar to discourage competition; include S&H of $1.25 for 3rd class, $1.75 for 1st. If you just want to copy memory, you can use XFER.BOOT (re. issue #16, #25, and #58) or a commercial card (6 are listed in issue #73) such as Snapshot, Crackshot, Wildcard (all three reviewed in issue #1), Alaska Card, Instant Replay, Copy Master, or Sr.Prom (perhaps James Heil, 3809 Cynthia Dr., Pittsburgh, PA 15227, still has one for sale or Mr. Beard didn't grab the one mentioned in issue #64). (I'm unfamiliar with Sr.Prom, Alaska, or Snapshot; the others are sometimes incompatible with various other cards and require extensive editing of the "captured" program to remove useless files—the entire memory is dumped, remember—and they don't work if the program accesses the disc again.)

The best deal of all, however, is "None of the above!" Jim Sather described how to create a "DOS HOSS" that loads DOS 3.3 from an Integer card in Understanding your Apple IIe, then developed it into a multiple program device called the QuikLoader, which is sold through the Southern California Research Group (SCRG, P.O. Box 593-R, Moorpark, CA 93020, 805-529-2082 or 800-736-1484, $179.50; a burner's another $149.50—have a Computist volunteer do it!). The price seems—formidable?—but it's still "worth it" if you use any program(s) repeatedly; it's much like taking that plunge when you buy a hard drive. You could have "instant-on" programs by using battery-backup RAM cards, but it'd cost twice as much (Memory+ doesn't list a Q-card anymore, so it'd be $340 for an RGB with Piggyback) and you don't get the other QuikLoader benefits: you press Z-RESET for DOS 3.3, Apple-RESET to use the disc's DOS, M-RESET to drop into Monitor then return to your program, or Q-RESET to select another program. They don't sell programs already "burned" anymore—the procedure's much easier than when I got mine so it's unnecessary—but they have ProDOS 1.1.1 ready to transfer from disc to an EPROM and they expect 1.8 "any day" from a third party. There are half a dozen sockets for EPROMs and you can mix 27064's with 27512's if you like, but Appleworks 3.0 alone uses three 27512's, they say, so plan on a few favorites. Better hurry, though; they deal only in Apple products and I haven't seen them mentioned in the ads lately!

To Zorro (issue #73, p15)—read the title page the next time you boot FrEd Writer; it is NOT public domain, but rather Freeware—you can circulate it "to other teachers" but you can't sell it. There are no restrictions on "public domain"—you can even claim it's your own if you're unscrupulous and think no one will notice—but technically (not likely, since they're pretty good guys) they could start charging for it at any time or even prosecute you for misrepresenting their product. (I'd never want any of my work listed that way; as Brian A. Troha (issue #71, p6) says, "Credit is my payment and nobody likes to get ripped off!") The last member of this trio is Shareware, which is usually accompanied by a message to "send me $25 if you decide to keep it" when you download it from a BBS; Glen Bredon used to market ProSEL this way until it became well known. And do you still want to buy a burner (no DOCs) for, say, $70? It can be rewired (1 connection) to burn 27128's, but I want to go for 27512's.

To Bill Jetzer (issue #72, p28) No, the gameport's capacitor-discharge method is too slow for analog-to-digital conversion from a tape. By the way, your DOS 3.3 Enhancements look splendid, but I need the same information for ProDOS; I agree with Jeff Root (issue #72, p12) that (except for game protection) "DOS 3.3 is dead!"

To Stephen M. Caraco (issue #71, p13)—I don't have a IIgs and have never submitted softkeys of the very few discs I have figured out because I forget what I changed, but I can tell HOW to pick the JSR's and JMP's to change: watch what the machine does from the moment the program starts ("boot code tracing"), and when it does something you don't like, you change it by trial and error until it behaves! Different people try different things—that's why there are so many softkeys—and sometimes you're lucky, while sometimes you have to rewrite the whole thing. Personally, I like putting a $60 (RTS) at the beginning of a suspect routine or changing SEC's to CLC's for starts, but the programmer usually thinks of that, so.... Be sure to check out the rest of the disc before patting yourself on the back; I've had friends inform me that some of my attempts didn't work after all. Learn machine language (ML—16-bit for the IIgs) and expect to spend hours on your first attempts. You'll find explicit clues for the IIgs in issue #71, p6 and issue #68, p27, and elsewhere.

To Carl D. Purdy (issue #71, p12)—I can't help you with poor Carmen—it could be timing or alignment of the drive, power supply, etc. or even an address check. As for your Wildcard copy, it doesn't matter how much room is available, but how much is used. First check the program length: Wildcard copies used to be limited to 64K. If you've changed it to ProDOS from DOS 3.3, check also that MAXFILES isn't changed in the original; if so, try adding .SYSTEM to the name to load it instead of BASIC.SYSTEM or change it back to DOS 3.3 so you can scrunch down to MAXFILES1. Next check for incompatible cards by removing everything you can and hoping it's not one of your built-in LASER functions. Now see if it's checking to "protect" you from running on the wrong machine, possibly via the Machine ID# at $BE98, or BLOADing into a part of memory "protected" under ProDOS. Finally, try using Wildcard in a different slot. (This problem is unlikely if you've been successful before, but there was once a similar problem with an ALS Z-80 card (CALL A.P.P.L.E., Nov '83) and "it couldn't hurt".) Your last alternative is to deprotect the disc completely—and send in your softkey.

To Dave Grenda (issue #72, p15)—See notes to Purdy on lengths.

To Ron Stankiewicz (issue #73, p5)—I can see five reasons for undocumented functions: (1) the author needed to access different parts of the game during development and didn't want to play the whole thing every time, (3) it may serve to authenticate authorship in a copyright litigation, (2) he wanted to take a break without rebooting (as in Sky Fox), (4) he tells only half his "beta-testers" about an alternative to see whether it improves the game or whether anyone discovers the feature on his own (as in Super Mario), or (5) he has a warped sense of humor (like me) or just wants to give his customer a little bonus (like the back of Printshop or Karateka). The undocumented feature in CAT.MAKER is mostly like (1); I used it during development and it probably won't help on a protected disc, but it's there to try if you want—just press (S)ave when asked for the ID-#.

To The Executioner (issue #68, p17)—What you want to do is save your words and definitions as a set of text files, probably by chapter of the text you're using, and recall them randomly. If the entries are short, use a random-access file and READ X$,R#, where R# is the word-number; otherwise, use a sequential file and load the whole list into matrices or a two-dimensional array. Place the number of entries at the beginning (in Record 0) or end each set with an easily identified symbol like "?". Separate the word and its definition with a dash (search each "string" for K$="—") or use commas and INPUT"";A$,B$. If this is too difficult (it really isn't, as I've done it myself), write again and I'll make you a program by next summer, but give more information on what it should do: print in lists, suggest multiple choices, give the definition and you supply the word instead, etc.

To Duane E. Spencer (issue #67, p12)—Yes, I use a Laser 3.5 with a IIe running under ProDOS. If you want to store DOS 3.3 on the 800K discs, you'll have to switch to AMDOS, UniDOS, etc.

To Tim Valuk (issue #67, p20)—the program counter keeps track of which address your program will get its next instruction from. You read it by encountering a "Break" (00) as the program runs or by pressing ctrl-E while you're in the monitor. Find out more in a book about Machine Language (ML) for the 6502 or 65C02 chip, such as 6502 Software Design by Leo J. Scanlon or a "SAM" selection at your local electronics store.

To Leonard R. Simon (issue #65, p8)—it sounds to me as if you may have exceeded DOS's limit of 105 files/volume. You can open a "large volume" to give room for all those CON. files and take over some sectors for the volume directory (Subtle Solutions's "Hard Byte Editor", $29.95 + $3 S&H, 314 S. Red Lion Terrace, Bear, DE 19701)—don't forget to mark them "used" in the VTOC—or simply continue on another volume. Your best move would be to convert your Sider entirely to ProDOS (Advanced Tech Services, P.O. Box 920413, Norcross, GA 30092, 404-441-3322, $59.95 + $2.50 S&H) to get faster access and NO limit on the number of files in subdirectories grouped by subject or whatever. (The main directory is still limited to 51, though.) You must tell them the model ("blue label, red LED" etc.) in order to get the proper chip, so call first. While you're at it, get ProSEL ($50, same source; there's no point in trying to contact Glen Bredon personally, it seems, as my friend and I have both written to him and never even received an acknowledgment) to manage the entire system—copying, lost file recovery, sorting directories (any way you like, not just alphabetically), etc. Once you're in ProDOS, you can open your directories like text files and print them out, make catalog files, or whatever else you wanted.

To John Windle (issue #65, p29) **and Keith** (issue #71, p12)—using $F8D0 directly places "F8D0" in $3A-B and is useless, unless you like seeing $F8D0 disassembled every time! Follow the directions in the BLISTER "docs" instead. The usual way to print out this code is (1) CALL-151 and $300LL etc. until you know which parts and how many L's (screens full) you want, (2) return to BASIC and turn on the printer, then (3) repeat step 1. If you want to do it in a program, WRITE it to /RAM or print on the screen in a fixed location and pick it up as a variable by resetting the pointers as my BLISTER program does. (BLISTER should appear about the same issue as this letter.)

To Gintana (issue #64, p8)—the point is that ProDOS is faster, accesses more files (you can have a disc FULL of nothing but directories or catalogs if you want), has more commands, and keeps track of files better. The last point is the important one. The file names are laid out like a school "outline", so if you need a FILE RUNFILE,S6,D1 just like DOS 3.3 or if it's in a SUBDIRECTORY ("subheading" of the outline) that is found in a DIRECTORY on a disc named DISC, you type -/DISC/DIRECTORY1/SUBDIRECTORY/FILE. (There's a way to abbreviate so you can just type -FILE, but this isn't a tutorial.) Also, you never notice it, but ProDOS uses a "Machine Language Interface" (MLI) so no matter how ProDOS is changed, the same information can always be found in the same place. You can BLOAD X in DOS 3.3 to find X's address at AA72-3 and the length at AA60-1, but this overwrites whatever else you had in memory and it works only on binary files; in ProDOS, you could VERIFY X or even UNLOCK X to get the type, etc. into the MLI, then even if it's not a binary file you can BLOAD X wherever you want it, examine it, and write the changes back to disc—one byte at a time, if you like!

It's not perfect—I hate the way it "protects" you from BLOADing into the Keyboard buffer at $200-FF, for instance—but ProDOS is REALLY improved and not just "New Improved" like a bar of soap or whatever. It used to have one fault that kept me from accepting it for over a year, and that was its "Launching" system—how you start the machine. Booting DOS 3.3 without a HELLO program just drops you into BASIC with a "PROGRAM NOT FOUND"; ProDOS had an idiotic STARTUP (read "HELLO") that essentially said "Guess the disc and program names and I'll let you run it." Now you can get a Launch System like ProSEL, QUARK, SQUIRT, or BYRD'S BETTER BYE (incorporated into Appleworks 3.0) that does away with such nonsense; you won't notice any difference with ProDOS until you try to do something that DOS 3.3 wouldn't let you do anyway, like BLOADing a text file or running a program from a different disc by giving the name without slot/drive numbers.

To Marc Batchelor (issue #72, p10)—When I was shopping for my first "personal" computer in '82, an IBM salesman told me "You can't afford it" instead of giving me the price; what got me was the snob was right! I wouldn't trust anyone who makes decisions by the size of my wallet even if he were giving his machines away with prizes, so if a Tandy salesman told me his machine is just like IBM's and that MS-DOS is stored on a chip, do you suppose I bothered distinguishing which one he was referring to? Besides, the TS-80 had a severe "keyboard bounce" problem; Atari had no bounce because there was no keyboard; and Commodore had no utilities and a flimsy card to add four slots where there had only been one—so when I found a machine that INVITED me to explore with eight slots and a built-in Monitor, I made a commitment. A couple of years later, IBM let their "PC Jr" die of neglect just about a month before I decided to buy one with a hard disc for little more than a new Sider alone, and I knew I was with the "good guys." Sure, I "rag" on IBM, but it's just human nature, I suppose: whenever it appears I made the right move I'm delighted (Ho, hum, IBM! Apple already has mice) and when it looks as if my huge investment will vaporize (Apple stuff is all sold through IBM outlets around here, you can't buy a IIgs or Mac by mail order, and our only Apple dealer "wouldn't recommend" that I buy a IIe) I get a little paranoid.

You seem to be comparing apples and bananas, or rather the top of one product-line with the bottom of the other. An 8-bit machine running at 33 Mhz, indeed! So what does either have to do with a "user-hateful environment"? I wouldn't expect that of either a II+ or an XT, but flying at "speeds of 33 Mhz" and multi-tasking under UNIX sounds pretty normal for a Mac, if you ask me. And what is there to "imagine" about not using BRUN?—typing one extra "-" to EXEC/RUN/BRUN isn't really that onerous! Blame Microsoft if either version of BASIC is inferior, and blame the people who put together MS-DOS if they left out or included some distinctive feature. It's surprising that the PATH function wasn't included in ProDOS, but IBM and Apple both sell computers like cars; PATH is probably being saved for next year's model. Before you write your own, save yourself some trouble and check the public domain where IBM seems to get most of its stuff: it's almost certainly out there, along with all those languages you wanted. When you find it, you can make it the first .SYSTEM on the disc or tack it onto ProDOS, whichever you prefer. The important point is that NONE of this is intrinsic to the machine itself; what's "under the hood" is all the same, just as long as you're talking the same requirements, and much of the hardware and software came from NEITHER party—Atari had the original 6502 chip (via Motorola), Commodore had sprites and sound envelopes before the IIgs, a "human-resources" group invented the menu-style used in the Mac, etc.

A 32-bit machine will always be faster than a 16-bit which is much faster than an 8-bit—if speed is what really matters, buy a Cray (designed on an Apple)! The only real difference between Apple and IBM is that Apple had slots and a Monitor and a friendly smile, while IBM had money and a lot of overconfidence, yet it's easy to see why Apple appealed to "the common man"—me. IBM had all those neat things years ago because there was nobody else to sell them to, not because of their own creativity. (They created EBCDIC for their mainframes, but I think that was just an ornery attempt to make them incompatible with everything else.)

Then Apple came along. "Try your hand at programming! It's easy, here's a big red manual to help, and the slots are for things that haven't even been invented yet but you can install them yourself when they come!" Even Applesoft wasn't built in, just in case you needed that limited 4K for a special program. You can't play with a mainframe, but you can do whatever you like with a "cheap imitation", and third party developers scared the bejeebers out of IBM with Visicalc, 80-column displays, Appleworks, etc. and made them enter the "home" market long before they were ready.

Then Apple betrayed all of us! Now it costs $600/year to be a developer for Apple, and the earliest Macs had neither slots nor Monitor because the public "wasn't smart enough" to use them. The red manual that came with the II told you MORE than you wanted to know about the insides of the machine; the latest one compliments you on buying a machine with cute pictures that's entirely menu-driven, can't be stopped to see what's happening, has no place (apparently) to access machine code, and barely matches the display of six other cheaper boxes—is it any wonder IBM no longer feels threatened?

I've never seen a company bent on self-destruction before, but it looks as if my investment's in real trouble this time! Mac's, IIgs', IIe's, III's, Porta-Mac's, and who-knows-what-else can't possibly outsell "buy this machine and it's compatible with everything", especially when they can say "we have everything anyone else does plus a world-wide reputation for doing the work for you." Trying to sell one machine to elementary schools, another to high schools, and none at all to colleges (along with the idea that Apple is for "education") just to boost sales is pure madness! It perpetuates the notion that Apples are toys and can't be "serious". (Besides, I know only one person who even considered "education" when she purchased her machine, and then it was the reason for buying now instead of later; it had no influence at all on the type.)

I've played on a Mac, but even I wonder whether it's any better than an IBM with "Windows". There are people in business (and business teachers) who honestly believe that only an "International Business Machine" can do the job and Apple is for play, yet Apple advertises how FUN it is to learn on a Mac and pretends the II's are made by a different company that doesn't advertise at all! It really pains me to admit that you're right, but the clones have everything an Apple has to offer. (I still refuse to say anything about IBM except they're overpriced.)

I'm sorry for the poor wording (issue #69, p8) that upset you last year, and probably again in this article too. Don't take any of this personally, as I really enjoy your articles (except when IBM comes up) and would miss them; I'm just negative because of Apple's horrible marketing "strategy". I suppose things aren't really all that gloomy—just recently I heard of a club for III-owners (PC Jr counterparts) and the II's aren't that far gone yet. I'll try to think of something nice to say next year, okay?

Dr Crack **France**

Softkey for...

Champions of Krynn.

SSI

This is I think the latest release from SSI of the AD&D series. It's the first volume of the Dragonlance series. This version No. 1.0 has only a few bugs, for example: in combat mode, one of the five dragon types is made up of four sprites but when he attacks only two out of four are up-dated so you end up looking at a pretty strange-looking dragon. It is a very good game anyway, with good graphics and a good quest.

The four disks can be copied with a fast copier and can be sector edited. The copy protection consists of a documentation check: the game asks you for a word from the Adventurer's Journal (AJ). This is quite annoying after a while because you have to get the AJ, look up the entry and then count up to nine (!) words.

After some time, I was so sick of it I decided to get rid of it, so I searched for one of the words in the list. After some searching I finally found it. Between each one was $00 which had to be a separation between each word so if I wanted anything to be correct then I would have to put $00 everywhere. I tried that, and it worked.

You may have noticed that you can insert any disk when the computer asks you for

disk A after you have chosen the 'begin adventuring' line, I did, so I searched every disk for the list and there it was, on every single disk.

Once you have made the changes you can answer anything, even 'return' to the question.

Boot disk

Trk	Sct	Byte	From	To	
18	0E	C6-FF	??	all to 00	
	18	0D	00-BE	??	all to 00

Disk A

Trk	Sct	Byte	From	To	
1B	03	C6-FF	??	00	
	1B	02	00-BE	??	00

Disk B

Trk	Sct	Byte	From	To	
19	0D	C6-FF	??	00	
	19	0C	00-BE	??	00

Disk C

Trk	Sct	Byte	From	To	
1A	05	C6-FF	??	00	
	1A	04	00-BE	??	00

Disk D

Trk	Sct	Byte	From	To	
1C	00	C6-FF	??	00	
	1C	0E	00-BE	??	00

Disk E

Trk	Sct	Byte	From	To	
1B	02	C6-FF	??	00	
	1B	01	00-BE	??	00

Disk F

Trk	Sct	Byte	From	To	
1B	04	C6-FF	??	00	
	1B	03	00-BE	??	00

? I have a question to ask all old Computist readers: Has anyone been able to crack Questron I? If not, has anybody been able to sector edit every track or sector?

I think that your magazine is the best, it's the only one I'm going to keep subscribing to even though most of the stuff is now for Apple IIgs.

Rod O'Brien — NY

This is my first letter to Computist, even though I have been a subscriber for a few years now. I am only beginning to understand most of the procedures listed in the magazine but even when I don't understand the reasons for a crack I still benefit from the use of many of those printed. I guess my guilt finally got the better of me and I am sending in what information I can to try and help some of those who have helped me.

Playing Tip for...

Dungeon Master

FTL

Dungeon Master Puzzles

To George Bigelow (issue #73, re. playing tips for Dungeon Master). The third level has a puzzle called "Cast your influence, cast your might" which George could not figure out. The answer to this puzzle is two part, first cast a spell of LO ZO to open the door, then have your strongest character throw an object into the next room, a club or stone will do. This trips the plate on the other side closing the pit.

Another good thing to do is have your Ninja character as the active hand when moving through the maze. As you move from area to area throw things in front of you. This allows the Ninja to gain points and levels without fighting and it doesn't matter what is thrown, rocks old swords from defeated enemies or even a shield. It all adds up.

If you want the complete set of maps and game hints for Dungeon Master contact:

Mullen Graphics
518 Roycroft Ave.
Long Beach, CA 90814

It is worth it just to not have to map out the dungeons.

Softkey for...

Dungeon Master

FTL

In Computist #70 there are two Softkeys for DM (page 12 and page 18) from Bob Thanski and Jim Ross. I used the Jim Ross softkey but the problem is that my copy of CopyII+ (v9.1) would only copy part of the tracks.

When reading the original the program stops a little more than half way (at track $34 on mine) and then prompts the user to insert the duplicate disk. When writing to the new disk it then writes in garbage for the tracks above the stop point. This will cause more than one error to appear when verifying the disk and will not allow the program to run.

The way around this problem is to follow Jim's softkey BUT to go back and then recopy the tracks from the stop point to the end of the disk again. Then follow the Block edit in the original softkey. This will get you a deprotected and workable copy.

Step-by-step

1. Use the Manual Sector Copier function of Copy II+ and ignore the error on Block $17.
2. Make a note of where the programs stops reading the tracks and asks for the copy to be inserted. Write it down.
3. Recopy the leftover parts to your copy from the stop point to the end of the disk. (Use the same copier program)
4. Manual Bit copy Track 0 with Sync = N and Keep = N
5. Verify copy to insure there is only one error (on block $17)
6. Sector edit block $104 (it was here that I found another difference in Jim Ross's Softkey).

Blk	Byte	From	To
104	110	18	38
	113	38	18

Each user should check for the correct bytes on their version of DM, mine was v2.0 of the game. You now have a cracked copy to play from.

Problems with the softkey

I have had the program freeze up and the screen go weird sometimes on the cracked version when I "save and play" the game. It does not affect your save so you can reboot and start from where you saved without a problem. The boot sequence is just so slow. A quick tip to avoid this problem is to "save and play" and then immediately throw an object or cast a spell (one that can be seen moving on the screen such as a poison bolt etc.) and see if there is movement. Your characters will be able to move all the time but sometimes objects and spells will be suspended in the air never to land and that prevents the character from ever being used again. The results: you will die sometime soon. If you experience this reboot the game and restore the game and you will find no problems. It just is a slow process.

? Maybe someone out there knows why this happens and can come up with a fix. Also, does anyone know how to install DM on a hard drive?

Copy II+ (9.0) and hard drives (with GS/OS)

I have been experiencing trouble with my Apple IIgs and Copy II+ (V9.0). It seems that this version has a bug in it that does not allow the GS to use GS/OS correctly. The shutdown sequence of GS/OS includes two messages, the first one asks if you want to really shut down etc. and then when you answer yes, it shows you the "you may safely shut off your GS now" message. This allows GS/OS to close all its tolls etc. (I think). The problem with Copy II+ (V9.0) is that when installed on your hard drive it does not allow GS/OS to shutdown properly. This happened regardless of whether or not Copy II+ was used during the session. When I removed it from my hard drive all problems with shutdown disappeared. A call to Central Point Software allowed me to inform of this problem (it appears to be a new one to them) and got me a new version of Copy II+ (V9.1) and the problem seems to be gone. I now have Copy II+ (V9.1) on my hard drive and GS/OS works fine. Call for your FREE replacement if you are a registered user. If you are not a registered user the manual alone is worth the cost of the program.

Deluxepaint II— A patch for the IIgs

In Computist #74 (page 6) User #601 asks about using Deluxe Paint with GS/OS V5.0.2. There is a patch available on line for this change and it comes from Jason Harper (who has programmed some great Public Domain and Shareware programs) that allows you to patch DP II. I'll reprint the patch here for those readers who don't have modems but still need the patch.

This patch only works with DeluxePaint II version 2.0 (the program file has a modification date of 30-Apr-87) or Version 2.01 (22-May-87).

Use a COPY only to try this patch.

1. Boot into BASIC and at the prompt, "]" type,

BLOAD DELUXEPAINT, T$B3, A$300, L1, B$19C12
PRINT PEEK (768)

At this point a number (0 or a 1) will appear. If it is a zero proceed, if it is a one do not continue. The one means the patch has already been applied or it is a different version of the program

POKE 768,1
BSAVE DELUXEPAINT, T$B3, A$300, L1, B$19C12

The patch is now applied.

To Rex Creekmur (issue #72), there is an easy way to copy Micro League Baseball, use Copy II+ (any version that lists the program, I found it on version 7.4) and that's all.

If you don't have that version use these parameters to copy Micro League Baseball:

```
Copy Track 0 - 22 and Keep
the Sync
T0-T22,sync
```

? If anyone out there knows how to crack the Micro League Manager's Disk or Stat's Disk please let us know.

Momma — CO

Softkey for...

Crosscountry USA

Didatech Software

Crosscountry USA is an excellent geography program that has been around for several years. I have tried numerous ways to back it up and have had no success. B. Dudley Brett's crack for Crosscountry Canada (#71, p.16-17) finally unlocked this disk. The protection is very similar, a matter of moving the protection a few bytes. For a more complete explanation, read his excellent article.

Step-by-step

1. Copy both sides with any copy program that ignores errors.
2. Use a sector editor to make the following changes:

Trk	Sct	Byte	From	To
$00	$04	$92	4C	00

3. Boot the disk you have created and wait for the '*'. Edit the hello file that is in memory.

Addr	Old value	New value
0892:	D0 EF	EA EA
OB30:	C8	EA

4. Then type:

UNLOCK COPYRIGHT 1985
BSAVE COPYRIGHT 1985, A$0805, L$373
LOCK COPYRIGHT 1985

Softkey for...

MECC 3.5" Disks (1990)

MECC

MECC is now putting a lot of their programs on 3.5" disks as well as 5.25". Some of the programs are designed solely for the GS. Unfortunately they all are copy-protected and the charge for the copy disk is outrageous. Anyone who has been in the classroom knows better than to put an original in the hands of students. After the last wait to get disks from the central office I decided to try to crack them. With the help of articles by several fellow Computists (in particular Brian Troha's article in Computist #65, p.33-34) I found the protection and removed it.

The current MECC disks are either 128K compatible or are written for a 1M GS. Each has a bad block, usually Block 7 or Block 8, and all the disks I tested deprotected in the same manner. In all cases you are changing a SEC 38 (which fails the carry check) to a CLC 18 (which clears the carry flag).

To find the protection code, I first copied the disk with a copy program that ignored errors. Then I searched the copied disk for either $20 00 BF 80 (on the 128K ProDOS 8 disks) or $22 A8 00 E1 22 00 (on the GS/OS 5.02 disks). The P8 disks all have the protection in the file MECC.SYSTEM. On those disks you should find code that looks something like this:

```
20 00 BF        JSR  $BF00
80                   read block
11 22
B0 98           BCS  $1174
18              CLC
60              RTS
2C E5 21        BIT  $21E5
30 91           BMI  $1174
38              SEC
60              RTS
```

The critical pattern to watch for is the CLC/RTS and SEC/RTS combinations. You want to change the SEC to CLC.

On GS/OS disks the code will look like this:

```
22 A8 00 E1     JSL  $E100A8
22 00                read block
...
22 A8 00 E1     JSL  $E100A8
22 00                read block
...
C9 27           CMP  #$27
00              BRK
D0 02           BNE  $11CF
18              CLC
60              RTS
38              SEC
60              RTS
```

Same CLC/RTS, SEC/RTS pattern as before. Again the change will be to change the 38 to an 18.

The following are the specific programs I tried and succeeded with:

Time Navigator Leaps Back

Blk	Byte	From	To
$28	$1DC	38	18

Murphy's Minerals

Blk	Byte	From	To
$28	$1F5	38	18

Sun and Seasons

Blk	Byte	From	To
$28	$1B5	38	18

Fossil Hunters

Blk	Byte	From	To
$D6	$1DC	38	18

Five-Star Forecast

Blk	Byte	From	To
$7	$1DC	38	18

Spelling Puzzles & Tests

Blk	Byte	From	To
$57	$1A3	38	18

Probability Lab

Blk	Byte	From	To
$28	$1DC	38	18

Estimation: Quick Solve I

Blk	Byte	From	To
$28	$1DC	38	18

Estimation: Quick Solve II

Blk	Byte	From	To
$28	$1DC	38	18

Designer Puzzles

Blk	Byte	From	To
$42A	$1CF	38	18

Designer Prints (1989)

Blk	Byte	From	To
$230	$CE	38	18

Mercury (1989)

Blk	Byte	From	To
$B4	$180	38	18

Softkey for...

Slide Shop (GS)

Broderbund

The protection on the GS version of Slide Shop was a bad block check on block 1599. I used the search for 20 00 BF 80 as suggested by Brian Troha and found it on Track $2B (43). There were three occurrences of those bytes on that track, so I found the beginning of the checking routine and replaced the first byte of the routine with an RTS (60).

Blk	Byte	From	To
$2B	$1BA	A9	60

Softkey for...

European Nations And Locations

Designware (Britannica)

Evidently they have added another layer of protection to this disk since James Harvey did his softkey (#74, p.21). The disk format has been changed by changing the data epilogues in some of the sectors. Fastcopy programs will no longer read the disk. Use COPYA and tell it to ignore the data epilogues. Then continue with Harvey's crack. This gave me a softkeyed copy.

Step-by-step

1. Boot your DOS 3.3 system disk.
2. Tell DOS to ignore epilog errors and use COPYA to copy the disk. (Alternative: Use COPYA+ and "Ignore" data epilogues.)

```
RUN COPYA
ctrl C                          at the menu
CALL -151                       enter the monitor
B925:18 60
3D0G                            return to BASIC
70                              to avoid reload of COPY.OBJ
RUN                             restart COPYA
```

3. Make the following sector edits to the copy.

Trk	Sct	Byte	From	To
$00	$03	$9C-A0	A9 38 8D 9E B9	EA EA EA EA EA

4. Write the changes back to your disk.

Rick Davis _____ **TX**

Softkey for...

Apple Panic

Broderbund

This softkey should work (with minor modifications) for most of the Broderbund software of a few years back.

1. Boot your Starter Kit disk. (or DOS 3.3 System Master).
2. Insert a blank disk in the drive and INIT a Slave disk.

```
INIT HELLO
DELETE HELLO
```

3. Enter the monitor.

```
CALL-151
9600<C600.C6FFM  Move Disk II code into RAM
```

4. Make Boot 0 exit to our code instead of the Boot 1 code at $0801. Make the correct patch for your computer.

```
96FA:98                         For II+, IIe only
96FD:98                         For IIgs only
```

5. Make the rest of the patches.

```
9801:2C E8 C0                   stop drive motor
:4C 59 FF                       exit to monitor
9600G                           load next boot stage
```

6. Boot 1 is now in memory. Move the Boot 1 code up so we can modify it.

```
9800<800.8FFM
```

7. Make Boot 1 move our code and then exit to our code instead of the Boot 2 code at $0301.

```
9805:98
9843:93
9301:2C E8 C0 4C 59 FF  Same as in step 5 above
9600G                   load next boot stage
```

8. Move Boot 2 up to where we can modify it.

```
9300<300.3FFM
9343:4C 00 90           Jump to our check code
9000:A5 3E C9 5D D0 03 4C 5D
9008:02 2C E8 C0 4C 59 FF
9600G
```

This code looks at the current pointer and checks it against $5D. If its not equal to $5D, it returns to Boot 2. If it is equal to $5D, it will stop the disk drive and go to RESET.

9. Here is where the minor modifications will be needed for different software. Type:

```
B700L
```

Look at the code until you see a JSR to some location. This is where the program will GOSUB to do the title page and unless you actually want to look at this every time you can ignore it. Apple Panic is a JSR $1000 at address $B745.

Continue looking at the code by typing a L <return> until you find a JMP to some location. Apple Panic has a JMP $4000 at address $B771.

Write the address down. Now modify the code at $9800 (Boot 1) with:

```
9844:2C E8 C0 4C 59 FF
```

Modify the code starting at $9000 with:

```
9809:A9 4C 8D 71 B7 A9 44
9810:8D 72 B7 A9 02 8D 73 B7
9818:4C 00 B7
```

9809:A9 4C	LDA	#$4C	Jump Instruction
980B:8D 71 B7	STA	$B771	Use address where JMP was found.
980E:A9 44	LDA	#$44	Low address of modified Boot 1 code to go to.
9810:8D 72 B7	STA	$B772	Address + 1
9813:A9 02	LDA	#$02	High address of modified Boot 1 code to go to.
9815:8D 73 B7	STA	$B773	Address +2
9818:4C 00 B7	JMP	$B700	Execute DOS code.

```
9600G                           run the changed code
```

10. Move game code to safe area.

```
2000<4000.93FF
```

11. Put Slave disk in and boot it with.

```
C600G
```

12. Save the game code.

```
BSAVE APPLE PANIC, A$2000, L$53FF
```

13. Load game code at it run location.

```
BLOAD APPLE PANIC, A$4000
```

14. Resave game code.

```
BSAVE APPLE PANIC, A$4000, L$53FF
```

Since Apple Panic is a single load program, it will run under DOS 3.3 or ProDOS.

Bitkey for...

War in the South Pacific

Strategic Simulations, Inc

Requirements:
Bit Copier (I used Copy II+)
two blank disks

A normal RDOS is on the back side of the disk. Boot Copy II Plus and use Manual Bit Copy. Copy the back side with no changes.

On the front side:

1. Change parameter $0B from 1 to 2. (After hitting return for what tracks to copy and it gives you the menu on the bottom, press the '/' key and press the 'B' key and then the '2' key. To copy just press return.)
2. On even tracks ($00, $02, $04, $06, $08, $0A, $0C, $0E, $10, $12, $14, $16, $18, $1A, $1C, $1E, $20, $22) just press the 'Q' key.

2a. On track $01:

Press the 'F' key, then type D4 AA 96.

Press the 'C' key and type D5.

Step 2a only has to be done once.

3. Now press the 'R' key to repeat find (when cursor is no longer on a D4 press 'Q' key)
4. Press the 'C' key to change a byte and then type 'D5' to change 'D4 to D5'.
5. Repeat this until all $22 tracks are copied.
6. Now copy tracks $00 and $01 from back side to front side.

There is an Editor program on the front side which is now available to change the game.

Softkey for...

Shiloh

SSI

Use same procedure as for War in the South Pacific, except do not copy the back side tracks $00 and $01 to the front side. (They discovered their mistake.)

Instead on the front side look at Track $00 Sector $0A. Place cursor on byte $3E. Press 'L' key to look at disassembly.

It should look like this. (If not it may have been moved so scan the disk for it)

093E:AD 78 BF	LDA	BF78	get track identifier
0941:C9 D4	CMP	#D4	was it D4?
0943:F0 03	BEQ	0948	If BF78 = D4 then branch
0945:EE 7B BF	INC	BF7B	Not equal to D4 so make it D5
0948:CE A7 BF	DEC	BFA7	subtract from # of sectors to read
094B:D0 E3	BNE	0930	Not all of them read if not zero
094D:60	RTS		All sectors read in so return

J L Walters has made a point of collecting the complete works of "Krakowicz", one of the more prolific writers on copy de-protection in the "early days" of Apple computing. Some of the material is dated but all of it is interesting. Beginners should read from start to finish, old hands can probably skim thru quickly. There are 22 parts in all. We'll print them, in order, in upcoming issues. My personal thanks to Mr. J L Walters for the time and effort he spent in putting together this collection and for sending it to us. Presented now are parts 1 and 2.....
.....RDEXed

Krakowicz _____

The Basics of Kracking (part 1) ROMs, PROMs and F8's

Along with a number of requests for material useful to those who are not yet in the ranks of professionals in this field, it has been pointed out to me that I am all too willing to suggest burning this PROM, installing that ROM, and generally making wholesale hardware changes in an unsuspecting Apple, without providing background information for the up-and-coming Krackists of the future.

This series, while aimed at the beginning to intermediate Krackist, will still assume a reasonable knowledge of assembly language. If you find these discussions are still too heavy into machine code for you, then it's best to buy a book like Roger Wagner's "Assembly Lines" or equivalent, and study it carefully (if, on the other hand, you find that this is all beneath you, just keep a knowing smirk on your lips as you skip lightly over these episodes—there might be something you missed because you had a bad hangover one day in Kracking 101).

In this and future episodes in the "Basics of Kracking" series, we'll deal with the fundamentals of the Krackist's art, starting with the how (and why) of making alterations in the Apple's "permanent" memory. First of all, the most important single tool available to the aspiring Krackist is replacing the autostart ROM on the mother board with an "old monitor" ROM. With this ROM in place, you can hit "reset" whenever you want, and always be returned to the monitor for the beginning of the snooping process. This change, incidentally, will make available to you a reasonable set of "step and trace" utilities (see the Apple II Reference Manual. pp 51-53).

To understand what the differences are between the two ROMS, let's take a minute to examine what pressing the "reset" key does (omigosh, Maude, there he goes again on that detailed technical crud!). Instead of going through the keyboard input routine at $C000, the reset key is connected directly to pin 40 of the 6502 microprocessor chip. When this pin is connected to ground (0 volts), the computer jumps unconditionally to the address contained in locations $FFFC and $FFFD. This is not a true interrupt, since the Apple forgets what it was doing before the line was "yanked," but it is an example of "vectoring" or sending the computer to a specific place by setting an address into the program counter. In the autostart ROM, these two locations contain $62 $FA, so the next instruction to be executed is at $FA62. This series of routines (see p. 143 and pp. 36-38 of the reference manual) checks to see if the computer is being powered up for the first time (coldstart) or reset with the power on (warmstart). If it is a warmstart, the system jumps to the instructions at locations $3F2 and $3F3, and begins running the program found there (usually basic at $E000). The "old monitor" ROM, however, has $59 $FF stored in $FFFC-D. This causes an Apple II (or a II+ with an integer card and the red switch "up") to go to routines which set up the keyboard for input, the TV for output, and wind up in the monitor with the "*" prompt displayed. In contrast to the AUTOSTART Rom, where anyone can tell the reset button where to go, there is no way to prevent a reset from going to $FF59 and winding up in the monitor. This is obviously essential if you want to break into a game and start examining the code, but it has its own set of problems.

In the process of setting up the I/O described above, especially in setting up the text window on the screen, a number of locations in zero page must be changed. The following locations will probably be altered (all hex): 20, 21, 22, 23, 24, 25, 28, 29, 32, 33, 35, 36, 37, 38, 39, and 48. Worse than that, the entire screen scrolls up one line when the monitor prompt is printed, which loses the entire top row of the text screen (locations $400-$427), and alters the contents of all the other locations from $400-$7FF, with the exception of the "scratch pad" regions at $478-$47F, $4F8-$4FF, etc. (The computer wimp at your school says that the top line "falls into the bit bucket," but you know how everyone feels about him.)

As most software protectors know, this will keep most of the amateurs out of the program, and you'll see evidence of this technique in the form of a lot of "garbage" on the text screen when you reset out of a protected game. Our job, then, is to keep these zero page and screen memory locations from being lost, since most protection schemes use these areas in some way or other (Broderbund, for example, has recently been storing the address marker for the disk track in locations $20, $21, and $22).

The safe way to prevent information from being lost from these "volatile" locations is to transfer all of the contents to a safe area—Locations $2000 & up (or $4000 & up) where a HI-RES picture normally resides. In fact, it would be best to save everything from $0 to $8FF, since booting a diskette to save the data also destroys locations $800-$8FF. (Remember the first law of disk kracking - track 0, sector 0 always starts with D5 AA 96 and always loads into $800-$8FF). Because this is the beginning class, let's look at two examples of short binary subroutines that will do the "save" for us. Both start, as will be explained later, at location $FECD in the F8 ROM. The first is the most straightforward and easiest to follow:

```
LDY  #$00     ;clear Y-register
LDA  $00,Y    ;Get a byte from 0+Y <— $FED0
STA  $2000,Y  ;store at 2000+Y
LDA  $0100,Y  ;then from 100+Y
STA  $2100,Y  ;to 2100+Y
LDA  $0200,Y  ;and so on until
STA  $2200,Y  ;we have covered
LDA  $0300,Y  ;all the memory
STA  $2300,Y  ;'pages' from 0 to 8
LDA  $0400,Y  ;and stored into
STA  $2400,Y  ;pages 20 to 28
LDA  $0500,Y
STA  $2500,Y
LDA  $0600,Y
LDA  $2600,Y
LDA  $0700,Y
```

```
STA  $2700,Y
LDA  $0800,Y
STA  $2800,Y
INY            ;then add 1 to Y-reg
BNE  $FED0     ;and repeat if < 256
JMP  $FF59     ;when we're all done jump to
                    monitor start
```

This 61-byte routine, if it could be executed automatically when the reset key is pressed, would safely stash all of the changeable memory and exit gracefully into the monitor.

A more compact and general, but less obvious, routine is shown below. It is included because it is typical of the "memory move programs" that we will eventually have to write in kracking almost any program.

```
LDY  #$00      ;clear Y-register
LDA  $00,Y     ;Xfer the zero page to <— $FED0
STA  $2000,Y   ;2000-20FF so we can use
INY            ;the zero page memory
BNE  $FED0     ;for the other moves
LDA  #$00      ;Set up locns 0 & 1 as a
STA  $00       ;2-byte pointer for the
STA  $02       ;source address, Use 2&3
LDA  #$01      ;as 2-byte pointer for
STA  $01       ;the destination address
LDA  #$21      ;starting at $2100
STA  $03
LDA  ($00) <-  ;Get a byte from 100-up
STA  ($02)  ^  ;store at 2100-up
INC  $02    ^  ;Increment lo-order byte
INC  $00    ^  ;of source & destination
BNE  ->->->^   ;(back to LDA ($00)
             ^ ;If lo-order is < 256
INC  $03    ^  ;If lo-order = 0, Inc the
INC  $01    ^  ;Hi byte of each
LDA  $01    ^  ;Check to see if hi-byte
CMP  $#09   ^  ;is 9 -We're thru at 8FF
BNE  ->->->^   ;If not, loop back to
               ;the load/store until
               ;we're all done
JMP  $FF59     ;Exit thru monitor
```

Unlike the first routine, this one (at 47 bytes) uses RAM locations 0 through 3, so the zero page must be transferred before it is altered by using those addresses as pointers. While the first routine must grow by six bytes for each additional page transferred, the second needs only to have the "9" in the compare statement changed to the appropriate value one higher than the last page number being transferred.

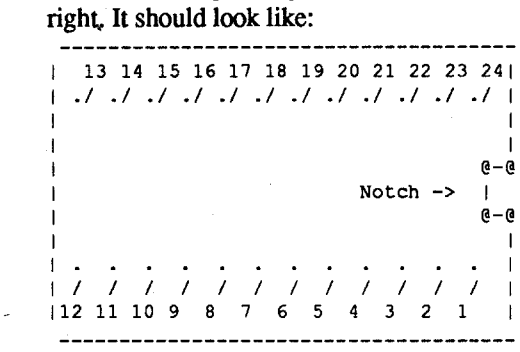
To return to the business of altering ROMs, it is easy to see that an autostart ROM could be made to behave like an old ROM just by changing locations $FFFC-D to $59 $FF from $62 $FA. (A note to the faint hearted: you can buy an old monitor F8 ROM for about $10 and plug it directly into your Apple's F8 socket, but you won't have all the benefits we've been talking about). As long as we're going to the effort of making a change, though, we might as well add one of the routines above and allow the new ROM to save the volatile memory for us. To do this, we'll have to give up something in the ROM, and the most easily surrendered area for most of us is the tape read/save routines at $FECD. If we then changed $FFFC-D to $CD $FE, the memory from $0 to $8FF would be saved to $2000-$28FF every time the "reset" key was pressed. Since it's sometimes inconvenient to have that happen when the reset key is pressed, we can require that a specific key also be pressed to make it occur. These few instructions inserted before either of the routines above will give a "reset and save" when the "-" key is held down (or was the last key pressed), while giving a regular "old reset" the rest of the time.

```
LDA  $C000     Look at the keyboard
ROL            Mask off high bit
CMP  #$5A      Was it "-"? ($2D X 2=$5A)
BNE  ->->->    If not, branch to the location with
               the "JUMP FF59" instruction at the
               end of the save subroutine.
```

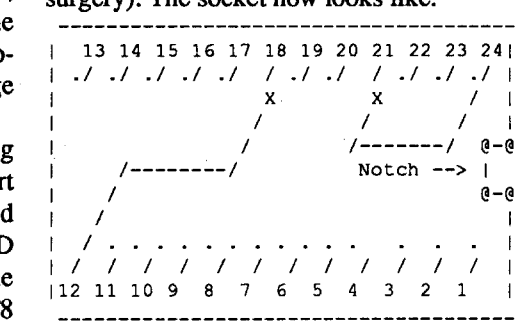
OK, OK - We all agree that these would be neat things to have in the F8 ROM, so how do we get it there? First, get hold of a PROM burner (PROMBLASTER, EPROM PROGRAMMER, etc.) that will program 2716 EPROMs. Each one is different, so I won't try to give detailed instructions on the actual programming. Buy or borrow a friend's old F8 ROM (or get the binary file) then type in or load in the changes you want to make at $FECD & up and at $FFFC-D, and program a 2716 EPROM with our modified version of Apple's F8 monitor ROM.

All that remains to take full advantage of the new F8 ROM is to make a slightly modified socket and plug it in. Both the 2716 and the original 9316 ROM used by Apple are read-only-memory devices holding 2K by 8 bits of information ("16K" ROMs), but the pinout, or assignment, of chip functions to pin numbers is slightly different. To use the 2716 in a board designed for a 9316, you need to tie pin 21 to 5 volts (pin 24) and tie pin 18 to ground (pin 12). You could modify the PROM itself, but you're liable to ruin the chip, and it creates a real magilla if you need to reprogram it. (A ROM card, such as an integer card, can be used for 2716's if two jumpers are connected at the top of the card, and ->only<- 2716's are used in all of its sockets after that).

Get a 24-pin, preferably low-profile IC socket and orient it with the pins up and the notch indicating the "pin one" end to the right. It should look like:



Using a low-wattage soldering iron, solder a short piece of 26-30 gauge wire between pins 21 and 24, and another one between pins 12 and 18. Make the connection as close to the socket as possible, and try to avoid getting any solder on the ends of pins 12 and 24. Cut off pins 21 and 18, again as close as possible to the socket. (Plugging another socket into the one being modified will help to prevent distortion during the surgery). The socket now looks like:



X = *No pin*

Double check the connections on the bottom of the socket, and plug the 2716 into the socket, being careful to match the notched end of the chip to the socket. Make sure that the power to the Apple is turned off, and plug the assembly into the F8 socket on the mother board with the notch toward the front (keyboard) end of the Apple. Cross your fingers and turn on the Apple. If there is no familiar "beep", or if the TV screen stays white, or if the system doesn't respond to the reset key, turn off the power and examine the chip and socket carefully to find the error. If black clouds of smoke roll out from the Apple, forget where you read this. Actually, the most common mistake of inserting the chip backwards is seldom harmful to it, but does lock up the Apple's bus. Remember that both the 2716 and the 9316 that you removed can be damaged by static electricity, so handle with care and don't scuff your feet on the cat.

The Basics of Kracking part 2
Single-Load Games, Starting Locations, and Obfuscation

The first in this series was straightforward, since the hardware reset is a necessity to begin kracking. After that, the path divides, and there are many ways to producing an unprotected version of a program. The path you follow is governed by three things: The kind of program, the type of protection employed, and your own personal style. (Style, by the way, is primarily the result of limitations.) Try to keep an open mind and develop as much versatility as possible). The easiest kind of program to deal with is the one that is seen less frequently every month: The "single-load" program or game. These are programs which are loaded in from disk only once, and then are run strictly from memory with no disk access. In the good old days, almost every game was like this, and removing protection was not that difficult. On the other hand, when you read something like Olaf Lubeck's challenge in track $17, sector $D of CANNONBALL BLITZ: "You'll never crack it", there's more satisfaction when you get to say "oh, yes I did!".

In order to become proficient at this and the techniques to be discussed in future episodes, you will have to get used to committing a very unnatural act: Interpreting assembler code with no comments or instructions to guide you. The disassembler (monitor "L" command) is a great help in this work, since it translates machine code into assembler mnemonics, but the real burden falls on the ingenuity of the krackist. There is no substitute for experience, and no one can teach you how to do it beyond pointing out some of the techniques we use and warning you about some of the tricks used to keep you from succeeding.

The philosophy of attack with these games is to find the starting location—the address which will always restart the game and then to save the game (program) as a normal DOS 3.3 binary file. As a simple example of a starting location, you probably already know that when you mess up with Apple's "FID" program, you can restart by typing "803G" from the monitor. At one time, before the publishers got smart, a starting location was likely to be a common, even number like $800, $C00, $4000, or $6000, and it's still worth checking these old favorites" in case you find a naive or lazy author. If these fail, we will have to begin the process of memory snooping. This is the introduction to the unglamorous activity that occupies most of the time of the dedicated krackist. As always, Inspector and Watson in ROM are highly recommended, since they make the process infinitely easier. What we are trying to do is directly locate the beginning address of the program, or to search back to it from something we can recognize.

Since many games begin by displaying a HI-RES "banner" or game screen, a good place to start looking is the series of instructions that set up the HI-RES screen (there is a discussion of this in the doc for Masterkey Plus, but they make a few too many assumptions). Apple's screen display, as you probably know, is set up by accessing some "soft switches". in hex, these are locations $C050 to $C057 (sorry, but if you're going to learn the gentle art of kracking, you'll have to become fluent in hexadecimal —we won't pull any punches when it comes to number systems). It doesn't matter what you do to these locations, as long as you make a reference, so the following instructions all establish graphics mode:

```
LDA $C050
BIT $C050
ROL $C050
STA $C050
CMP $C050
EOR $C050
```

(Also, this one: LDY #$71 then LDA $BFAF,Y)

Many authors have established the habit, however, of writing the sequence:

```
LDA  $C054   Select primary page
LDA  $C057   Select Hi-res graphics
LDA  $C050   Select graphics mode
```

and sometimes,

```
LDA  $C052   Pure graphics screen
```

To find these instructions, use the Inspector's "find" function, and program it to search for the two-byte sequences of "50 C0" and "57 C0". Generally, as long as the writers aren't deliberately trying to confuse you, you will find one to several locations where these sequences are close to each other. You will also find some addresses that don't really contain a screen reference, since the search is only for two bytes (for you trivia statistics buffs out there, a given two-byte sequence would occur less that once in the entire RAM memory space from $0 to $BFFF if the distribution were truly random. It's not.).

To see if each occurrence of the pattern is the starting location, look backwards until you find an absolute end for the previous subroutine such as "RTS" or "JMP". Your subroutine should begin immediately after that, and you should assume for the moment that it's the starting location. If, for example, the location you found is $4123, test it by reloading the game, resetting it, and typing "4123G". If it runs, sit back and gloat, otherwise read on (it sounds unnecessary to reload, but the Inspector uses a few locations in pages 0, 2, and 3, so it's best to be safe). If Murphy's law of dynamic negatives is with you and the game didn't start, it's usually because you haven't found the true starting location. You then need to trace back further in the program sequence to find the real start.

There are three ways for another routine to get to the one you're looking at: JMP, JSR, and the family of branch instructions. To eliminate the third possibility, keep in mind that branches can reach up to $7F (127) locations away from either direction. this is equal to about 60 instructions, so you should review about one full page of disassembly printout (three screens full) before and rarely after what looked like a possible start. If you find a "BNE $4123", or "BCC $4123", etc., you will have to track back to the beginning of that routine and try again. Repeat this process until you find a location that can only be reached by a JMP or JSR.

To find out how the program got to this location, do a 3-byte search with the Inspector for a JSR $4123: 20 23 41. If nothing shows up, try the JMP $4123: 4C 23 41. One of these must produce a reference, or you messed up the earlier check for branches. Once you find the earlier reference, go through the same procedure to find the start of this routine, and try it out as a starting location for the game. If it doesn't work, try one more step further back (Krakowicz's fourth law of kracking says that if you have to go back more than two steps, you're probably not on the right trail).

A number of games still do us the favor of putting up a screen, perhaps playing a little music, and then waiting for the space bar or other key to be pressed. If it's not possible to find the screen setup, we still have a fairly obvious "hook" into finding the starting address, and in many cases the game can be saved "as is" by using the keyboard routine as the starting address. Don't worry for now about exactly how we will "save the game." we'll go through that carefully and thoroughly in the next episode.

Since the keyboard address is $C000, we can usually locate all the inputs by searching for the 3-byte sequence of "AD 00 C0" with the Inspector. Occasionally, the X or Y register is used to load keyboard data, so the sequences AC 00 C0 and AE 00 C0 should be tried if the first comes up blank (only the real bast—ds like Sirius use LDY #$67; LDA $BF99,Y for the keyboard input). Also, keep in mind that all the addresses from $C000 to $C00F will access the keyboard, and if someone was really determined to confuse you they could use $C007 one time, $C00D the next, and so on. If you know that the game uses the keyboard and the preliminary searches don't show how, keep on looking for these addresses, or the Sirius-type computed addresses. It probably means they have something to hide, and locating the keyboard read will reveal enough to make the search worthwhile.

If the program is waiting for the space bar, you will usually find a sequence like:

```
78E0    LDA  $C000   Read the keyboard
      * CMP  #$A0    Was it space?
      * BNE  $78E0   Nope, keep trying
        JMP  $6012   Yes, go to start
```

*These two lines are eliminated if pressing any key will start the game.

To check out $6012 as a starting address, set up to view the HI-RES screen (otherwise the game might be running while you watch a blank text screen) with: $C050 (CR) $C057 (CR), then type 6012G. As before, you will know at once if you were successful.

Another way to find a restart point is to search through the keyboard input routines for a restart key. It has become conventional to use CTRL-R as the restart command (occasionally CTRL-S or CTRL-B), and this is even easier to trace. In one of the routines following a $C000 reference, you will find a CMP #$92 (see the reference manual, p. 7 for the hex values of the keyboard). The location branched to or jumped to by a successful compare will be the restart for the game. Again, you can save the game as is and use your new-found starting location.

If these relatively simple approaches fail, you'll have to resort to the real grunt type of detective work—looking for something promising (we'll discuss boot-tracing as an alternative way of getting to this point in another episode devoted entirely to that technique). Likely things to look for are "setups", where a lot of zero page locations are initialized to begin the game:

```
LDA  #$00
STA  $23
STA  $57
LDA  #$12
STA  $30
LDA  #$E9
STA  $72
etc.
```

Or, sometimes, a game start is indicated by a subroutine sequence which maps out the path for the game (this is an indication of an experienced, well-disciplined programmer and thus is more commonly seen in business or professional programs, rarely in game programming).

```
JSR  $8CD
JSR  $CE4
JSR  $2020
JSR  $203D
JSR  $8FE
etc.
```

And, although it's less often the start of a program or game, a "jump table" can be a significant clue to the organization of the program:

```
JMP  $204D
JMP  $2433
JMP  $EF2
JMP  $2077
etc.
```

Unfortunately, snooping for these is a time-consuming, hit-and-miss operation, the real starting address can be anywhere from $0000 to $BFFF (or even via a basic subroutine in $D000-$F7FF, but I don't want to discourage you yet).

While it will be disconcerting to the beginner, as you get more experience you begin to enjoy defeating various deliberate attempts to throw you off the trail—the general subject of obfuscation, or intentional lack of clarity. Because the major software companies know we're out here waiting for their latest output, they often try to misdirect us or find innovative ways of hiding sensitive portions of the program with a variety of techniques. Take a look at the following piece of code from On-Line's Cannonball Blitz:

```
59E4:CE E7 59   DEC  $59E7
59E7:CF         ???
59E8:EA         NOP
59E9:59 EF EA   EOR  $EAEF,Y
59EC:59 AD 51   EOR  $51AD,Y
59EF:C0 AD      CPY  #$AD
59F1:54         ???
59F2:C0 AD      CPY  #$AD
59F4:57         ???
59F5:C0 AD      CPY  #$AD
59F7:52         ???
59F8:C0 20      CPY  #$20
59FA:60         RTS
59FB:5B         ???
59FC:20 C5 5B   JSR  $5BC5
59FF:20 4E 5B   JSR  $5B4E
```

This is an example of "self-modifying code"-instructions that change as the program is run. It's dangerous and generally poor programming practice, but it can be used to throw the dogs off the scent. At first glance, it looks like data or garbage stuck in before some real code. Let's look at exactly how it works. Executing the first instruction changes the second instruction from junk into a legal instruction:

```
59E4:CE E7 59   DEC  $59E7
59E7:CE EA 59   DEC  $59EA
59EA:EF         ???
59EB:EA         NOP
59EC:59 AD 51   EOR  $51AD,Y
59EF:C0 AD      CPY  #$AD
```

(If you have an old monitor ROM, you can type 59E4S to execute the first instruction). If we execute the second instruction, the entire picture changes:

```
59E4:CE E7 59   DEC  $59E7
59E7:CE EA 59   DEC  $59EA
59EA:EE EA 59   INC  $59EA
59ED:AD 51 C0   LDA  $C051
59F0:AD 54 C0   LDA  $C054
59F3:AD 57 C0   LDA  $C057
59F6:AD 52 C0   LDA  $C052
59F9:20 60 5B   JSR  $5B60
59FC:20 C5 5B   JSR  $5BC5
59FF:20 4E 5B   JSR  $5B4E
5A02:A9 04      LDA  #$04
5A04:8D EC B7   STA  $B7EC
5A07:A9 00      LDA  #$00
5A09:8D EB B7   STA  $B7EB
5A0C:A9 00      LDA  #$00
5A0E:8D F0 B7   STA  $B7F0
5A11:A9 60      LDA  #$60
5A13:8D F1 B7   STA  $B7F1
5A16:A9 40      LDA  #$40
5A18:20 45 5A   JSR  $5A45
5A1B:10 01      BPL  $5A1E
5A1D:A9 20      LDA  #$20
5A1F:91 5A      STA  ($5A),Y
5A21:AD 50 C0   LDA  $C050
5A24:A9 09      LDA  #$09
```

Suddenly, the screen setup code that was always there pops into view. This points out the value of searching with the Inspector, since even the closest scrutiny would probably not have made you suspect what was actually here. Notice, too, that the third instruction increments $59EA, so once it's been run, it's obscured again.

Another standard trick, also shown in this example, is called "false disassembly", and is dear to Edu-Ware, On-Line, IDSI, and Scientific Research Associates. Here, extra bytes are added for the sole purpose of giving a false indication of program flow; the fake bytes are then branched around. Look closely at the instruction in $5A1B-it says BPL $5A1E. The next instructions in sequence appear to the casual eye to be LDA $#20; STA ($5A),Y. Actually, the next instruction is JSR $5A91. This is crucial, since this subroutine loads in the game and does a nibble count. To see a whole bunch of false disassemblies in a row, look at the code in the actual subroutine:

```
5A91:A9 00      LDA  #$00
5A93:10 01      BPL  $5A96
5A95:20 A8 59   JSR  $59A8
5A98:00         BRK
5A99:27         ???
5A9A:C8         INY
5A9B:D0 FA      BNE  $5A97
5A9D:85 10      STA  $10
5A9F:F0 01      BEQ  $5AA2
5AA1:A9 A9      LDA  #$A9
5AA3:20 59 00   JSR  $0059
5AA6:27         ???
5AA7:C8         INY
5AA8:C8         INY
5AA9:D0 F9      BNE  $5AA4
5AAB:85 11      STA  $11
5AAD:49 B7      EOR  #$B7
5AAF:48         PHA
5AB0:A5 10      LDA  $10
5AB2:49 11      EOR  #$11
5AB4:48         PHA
5AB5:D0 01      BNE  $5AB8
5AB7:4C 60 08   JMP  $0860
5ABA:60         RTS
```

I strongly urge you to sit down and figure out exactly what the real program is here, and if possible, what it does. Cover up the explanation below, and go through the code byte by byte to eliminate the fake bytes. It's not just character-building—if you go through a few of these, you'll learn to recognize them when they pop up.

Those of you who really went through it, give yourselves four kracking honor points. For the rest of you, here's a listing of the functional equivalent (some addresses are changed because the junk bytes have been taken out):

```
5A91:A9 00      LDA  #$00
5A93:A8         TAY
5A94:59 00 27   EOR  $2700,Y
5A97:C8         INY
5A98:D0 FA      BNE  $5A94
5A9A:85 10      STA  $10
5A9C:A9 20      LDA  #$20
5A9E:59 00 27   EOR  $2700,Y
5AA1:C8         INY
5AA2:C8         INY
5AA3:D0 F9      BNE  $5A9E
5AA5:85 11      STA  $11
5AA7:45 B7      EOR  $B7
5AA9:48         PHA
5AAA:A5 10      LDA  $10
5AAC:49 11      EOR  #$11
5AAE:48         PHA
5AAF:60         RTS
```

This is also valuable because it introduces the concept of "jumping through the stack. The RTS instruction transfers the two bytes above the stack pointer in page one to the program counter, increments the low byte by one, and jumps to that location. Ordinarily, the bytes on the stack were placed there as a return address by the JSR instruction. In this case, in very roundabout fashion, the On-Liners have pushed two bytes on the stack and executed an RTS, which jumps to the location one higher that the values stored. The story of the subroutine goes like this: create a checksum by exclusive-oring together all the bytes from $2700 to $27FF, and store it in $10. This allows a check to see if any of the bytes in the nibble count routine were altered. Do a second checksum on every other byte from $2700 to $27FF, starting with a value of #$20. Store this in $11, then exclusive-or it with #$B7 to produce the low byte of the return address: $FF. Push this on the stack, exclusive-or the first checksum with #$11 to produce the return high byte of $26, then do the RTS to jump to $2700. When you look at $2700, you find this:

```
2700:CE 03 27   DEC  $2703
2703:EF         ???
2704:03         ???
2705:27         ???
2706:AD 24 27   LDA  $2724
2709:49 8A      EOR  #$8A
270B:D0 01      BNE  $270E
270D:20 8D 24   JSR  $248D
2710:27         ???
2711:D0 01      BNE  $2714
2713:4C A0 25   JMP  $25A0
2716:98         TYA
2717:59 00 27   EOR  $2700,Y
271A:99 00 27   STA  $2700,Y
271D:C8         INY
271E:D0 F6      BNE  $2716
```

(You see, now that we're familiar with this kind of trick, there's nothing to decoding that mess, is there?)

Stay tuned to Computist for part 3, when we finish this subject by answering the burning question "what is the window-shade technique?", and proceed to a discussion of memory moving and file saving.

George Sabeh — PA

Bitkey for...

The Usurper: Mines of Qyntarr

Sir Tech

Requirements:
Copy II Plus

This program is an all text adventure game with an excellent parser. I have been trying to copy it for the past year but have not succeeded. It will not copy with any of the available copy programs including EDD with card. It uses a DOS that has been modified drastically. Looking at the disk with a sector editor I discovered that track 0, sector 0-9 are in normal format. Track 0, sector A-F and track 1-22 are unreadable. A bit copy will not run. Breaking out of the program with the Senior Prom NMI and looking at the memory reveals the RWTS is in normal location at B800 and can be captured. The program can be converted to normal format but will not run with normal DOS. Accordingly we have to convert the whole disk from track $00-22 to normal and sector edit the markers to be able to make a functioning copy. All prologues and epilogues are changed except for track 0 sector 0-9.

To be able to make a bit copy and eventually unprotect the program, we need to use the captured RWTS to copy track 1-22, then modify the controller to copy track 0, sector A-F, and use a sector editor or a copy program that ignores errors to copy track 0, sector 0-9. The information required to achieve all this was discovered by snooping through memory to locate the Headers and Trailers used by the game and making modifications to Copy II Plus for the bit copy and to Super IOB with Newswap controller to unprotect. The information needed consisted of the following:

	changed	normal
Address Prologue	99 C9 A5	D5 AA 96
Address Epilogue	C9 CC EB	DE AA EB
Data Prologue	E4 A4 D4	D5 AA AD
Data epilogue	C9 CC EB	DE AA EB

Step-by-step

1. Boot Copy II Plus and enter the bit copy section.
2. Use bit copy option to copy track 0.
3. Use sector copy option to copy track 1-22 with following parameter changes:

57=99, 58=C9, 59=A5, 5C=C9, 5D=CC, 61=E4, 62=A4, 63=D4, 66=C9, 67=CC

This allows the sector copy program to read the disk properly and will produce a bit copy that will boot and run normally.

Softkey for...

The Usurper: Mines of Qyntarr

Sir Tech

Requirements:
Super IOB with Newswap controller
One blank disk
Sector editor
A way into the monitor is desirable
A fast copier that ignores errors may be helpful

To completely unprotect you need to be able to interrupt the program and capture the RWTS. I used the Senior Prom. Hitting Control-Reset will also work most of the time. I am sure someone who is proficient in writing controllers can accomplish the whole procedure including the sector edits and capturing the normal part of track 0 all in one controller. I am unable to do this and will describe a round about way of doing this. Maybe someone else will use this information to write a controller to accomplish all this.

Step-by-step

1. Boot the original and when you start playing interrupt with NMI or Control-Reset. Check to make sure the RWTS is still intact at B800 by listing the code at B800.
2. Move the RWTS to $1900.

1900<B800.BFFFM

3. Save the RWTS to your Super IOB disk. Use the name "RWTS.XXX" — same as the name in the Newswap controller line 10010.

BSAVE RWTS.XXX, A$1900, L$800

4. Format a disk with normal DOS.
5. Copy track 0, sector 0-9 by either allowing a fast copier that ignore errors to do this or use the sector editor to read one sector at a time and writing to the copy. Although this may take a couple of minutes, it is much simpler for a beginner.
6. Next step involves capturing the main program from track 1-22 using the Super

IOB. Boot the Super IOB disk and use the Newswap controller. When it asks if you want the disk formatted, hit control-reset to enter BASIC and get ']' prompt. Now list line 1010 and you should see the following "1010 TK=3:LT=35:ST=15: LS=15:CD=WR:FAST=1". This tells the copy program which track to start at and which sector etc. Change the TK=3 to TK=1 to have it start copying from track 1 to 22. Now while the Super IOB disk is still in the drive type RUN and press return. The disk drive will come on for a moment and you are back in the copy program. Proceed to make a copy and tell it NOT to format the disk when it asks.

7. The next step is to capture track 0, sector A-F. Run Super IOB with Newswap controller and break out of the program as in step 6. List line 1010 and this time change to the following 1010 TK=0:LT=0:ST=15:LS=9:CD=WR:FAST=1 and press return. Then type RUN and press return. Now proceed to make the copy. Now we have converted the whole disk to normal format.
8. The only thing left is to tell the original DOS that we are using normal markers. Use your sector editor and do the following sector edits:

Trk	Sct	Byte	From	To
$00	$03	$55	99	D5
$00	$03	$5F	C9	AA
$00	$03	$6A	A5	96
$00	$03	$91	C9	DE
$00	$03	$9B	CC	AA
$00	$02	$E7	E4	D5
$00	$02	$F1	A4	AA
$00	$02	$FC	D4	AD
$00	$03	$35	C9	DE
$00	$03	$3F	CC	AA

This completes the normalizing of the disk and produces an unprotected version that uses the original DOS. I would love to see one of the experts tackle the job of putting all this together into one controller. This would be of great educational value to me and will help me in the future to simplify writing controllers. I hope I included all the information needed to do this job. How about it experts?

Softkey for...

Typewriter

Power Up

Requirements:
Blank disk
Any whole disk copier such as COPYA, Locksmith Fast Copy etc.
Sector editor

This program is dated 1985. It allows you to use your computer as a typewriter. It prints one line at a time. This program is written in Pascal and is COPYA-able. This indicates that the format is normal and the disk uses a nibble count for protection. A working copy can be made using Essential Data Duplicator but requires several tries to make a reliable copy. Because of this I felt it best to remove the protection completely and free the program.

Step-by-step

1. Make a copy of the disk with COPYA. This copy will not run because of the nibble count.
2. Sector edit the copy to bypass the nibble count by putting a jump over it. If the protection has been moved on your disk, then scan for AD E9 C0. You will find this in only one location on the disk.

Trk	Sct	Byte	From	To
$0D	$08	$3E-40	AD E9 C0	18 90 4C

Bitkey for...

Prince of Persia

?

Requirements:
EDD v4

I agree with several of the readers in Computist 75 who wrote about this program. The format is very difficult to unprotect and is similar to Wings of Fury. It was suggested that a bit copy can be made using Copy II Plus. I was unable to accomplish this, but I can make reliable copies using EDD v4 using Preanalyze 00=90 or 00=B0. Using any other method will produce a copy that will run normally for the first two levels only. When asked to turn the disk over for the rest of the levels the program crashes. The method I used will produce a reliable copy and I can assure your readers of this since I have completed the game. Also I would like to say that this is one of the best games for the Apple II to be published for a long time.

Softkey for...

Star Rank Boxing II

Gamestar/Activision

Requirements:
Original disk
Blank disk
Sector editor
Whole disk copier such COPYA or Locksmith fast copy

This disk copies without errors with any copier. The format is normal and scanning the disk for the usual Activision protection code reveals this to be present on track 00, sector 9 starting at byte 12. Trying to bypass the protection with the usual methods described in previous Computist issues for Activision does not work. Apparently they have been reading Computist and left the protection intact, but added a different form of protection. The new protection can be found on track 1, sector 5 and 8. A bit copy can be made by using Essential Data Duplicator v 4 by using Preanalyze 00=B0.

I discovered the protection when I scanned the disk for the error message displayed when the copy is booted. The screen displays "PLEASE PUT ORIGINAL IN DRIVE 1". This code was found in two locations. This is the protection code that does a disk check if not original and I found it can be bypassed by simply putting a "60" or return at its beginning. If your copy has the code in a different location, then scan for 2C 54 C0 2C and change the first 2C to 60.

Step-by-step

1. Make a copy of the disk with any whole disk copier.
2. Sector edit the copy

Trk	Sct	Byte	From	To
$00	$05	54	2C	60
	$08	39	2C	60

This should bypass the protection completely.

Edward Eastman NE

Attention all Readers, Computist is looking for any small programs you may have developed for your own use. I chatted with Computist a while back and we discussed many topics. One direction mentioned for the magazine was expanding into programs for other readers. Any utility or game or programming caveat is welcome, so long as it is relatively small. However, let Computist determined if it is too large. So if you have a nifty one liner or neat algorithm send it in to help your fellow enthusiasts. Having had advance warning, here is my donation to the cause.

Epson Label Printer

This little gem was inspired by a box of one inch labels that I acquired quite a while ago. I found them handy to use as disk, address and return address labels. Unfortunately, I had to fire up my word processor just to put text on one. Plus I was constantly looking up control codes for the formatting I wanted. As I always do, I thought 'there has to be an easier way'. Searching through old magazines I found a program that printed labels for an Apple DMP, but I have an Epson.

All hope was not lost because I am a decent BASIC programmer. Browsing through my printer manual, I found all the information I needed. I discovered how to access my printer's special modes as well. This is my best BASIC effort yet. You will have access to condensed, draft, italics, NLQ if your printer has it, and be able to print six or eight lines per inch (label).

Using the program

First, when you enter a control letter into a label, it will appear as a blank until you press 'return', then it will turn inverse on the screen. Second all control characters are accepted but only the ones listed will be converted into printer control codes, that way you can still send special codes without modifying the program. Third, while editing, the left arrow will work as a delete key, the other arrows are considered control characters. And fourth, I have a parallel interface card in slot 1 of my 'Apple', you may have to change the printer routine at 6000 to match your system.

I have tried to make this program as user friendly as possible and still have it easily modifiable. All control characters to printer codes are in the subroutine at the end of the program. You can have any control character be any printer code you want to add. If you want to modify the program, but cannot follow it, contact me for a completely remarked version.

Epson Label Maker

```
10 REM 1" BY 3.5"
20 REM LABEL MAKER
30 REM FOR AN EPSON
40 REM CREATED BY
50 REM EDWARD L. EASTMAN
60 REM OCT 1990
100 HOME : INVERSE : HTAB 10:
  PRINT "◊EPSON◊LABEL◊MAKER◊" :
  NORMAL : PRINT
110 PRINT "◊THIS◊UTILITY◊IS◊DESIGN
  ED◊FOR◊EASE◊OF◊◊USE.◊USE◊THE◊FOL
  LOWING ◊KEYS◊FOR◊THE" : PRINT
  "CORRESPONDING◊EFFECT◊ON◊YOUR◊
  EPSON." : PRINT
120 PRINT "◊◊KEY◊◊◊◊EFFECT" :
  PRINT "CTRL-S◊◊SMALL◊PRINT◊(17◊
  CPI)" : PRINT "CTRL-L◊◊NLQ◊(10◊
  CPI◊IF◊SUPPORTED" : PRINT "CTRL-
  D◊◊DOUBLE◊WIDE◊(5◊CPI)"
130 PRINT "CTRL-I◊◊ITALICS◊(ALL
  ◊CPI)" : PRINT "CTRL-B◊◊BOLD◊
  (NOT◊NLQ)" : PRINT "CTRL-◊◊UNDER
  LINE◊START"
140 PRINT "CTRL-N◊◊RESETS◊DRAFT◊
  MODE◊(10◊CPI)◊AND" : HTAB 10:
  PRINT "UNDOES◊ALL◊OTHER◊MODES" :
  PRINT
200 PRINT "◊YOU◊SHOULD◊ALWAYS◊LEAVE
  ◊THE◊LAST◊LINE◊◊BLANK◊TO◊ALLOW
  ◊FOR◊THE◊SPACE◊BETWEEN" : PRINT
  "LABELS.
210 PRINT "◊YOU◊SHOULD◊ALSO◊KEEP
  ◊IN◊MIND◊THAT◊SOME◊MODES◊OVER
  RIDE◊OTHERS,◊CONSULT◊YOUR" :
  PRINT "PRINTER'S◊MANUAL"
220 HTAB 8: PRINT "PRESS◊ANY◊KEY◊
  TO◊CONTINUE" ;: GET A$: PRINT
230 HOME : INVERSE : HTAB 10:
  PRINT "◊EPSON◊LABEL◊MAKER◊" :
  NORMAL : PRINT
240 PRINT "◊SELECT◊LINES◊PER◊INCH
  ◊(6◊OR◊8):◊" ;: GET A$:L = 1:LI
  = 6: IF A$ = "8" THEN LI = 8
250 PRINT LI: PRINT
999 REM MAIN LOOP
1000 VTAB 20: PRINT : PRINT "A/
  Z◊TO◊CHANGE◊LINE◊NUM.◊OR◊'P'◊TO
  ◊PRINT" : PRINT "'ESC'◊TO◊EXIT◊
  OR◊'RTN'◊TO◊EDIT◊LINE◊#" L;: GET
  A$
1010 IF A$ = "A" OR A$ = CHR$ (97)
  THEN L = L - 1: IF L < 1 THEN L
  = 1: GOTO 1000
1020 IF A$ = "Z" OR A$ = CHR$
  (122) THEN L = L + 1: IF L > LI
  THEN L = LI: GOTO 1000
1030 IF A$ = CHR$ (13) THEN 2000
1040 IF A$ = "P" OR A$ = CHR$
  (112) THEN 6000
1050 IF A$ = CHR$ (27) THEN PRINT
  : GOTO 6100
1060 GOTO 1000
1999 REM EDITING ROUTINE
2000 VTAB (L + 2) * 2 - 1: HTAB (
  LEN (Q$(L)) + 1): GET A$
2010 IF A$ = CHR$ (13) THEN PRINT
  "◊" : GOTO 3000
2020 IF A$ = CHR$ (8) THEN IF LEN
  (Q$(L)) = 1 THEN Q$(L) = ""
2030 IF A$ = CHR$ (8) THEN A$ = ""
  : IF LEN (Q$(L)) > 0 THEN Q$(L)
  = LEFT$ (Q$(L), LEN (Q$(L)) -
  1)
2040 Q$(L) = Q$(L) + A$: PRINT A$;
  "◊" : GOTO 2000
2999 REM PRINT STRING ON SCREEN
3000 IF Q$(L) = "" THEN Q$(L) =
  "◊"
3010 VTAB (L + 2) * 2 - 1: HTAB
  1:LW(L) = 0: FOR X = 1 TO LEN
  (Q$(L)):A$ = MID$ (Q$(L),X,1):
  IF A$ < CHR$ (32) THEN GOSUB
  5000:A$ = CHR$ ( ASC (A$) +
  64): INVERSE
3020 PRINT A$;:LW(L) = LW(L) +
  (100 * (D = 1) + 41 * (S = 0) +
  59) * ( PEEK (50) = 255):
  NORMAL : NEXT
3030 PRINT "=" LW(L) / 1000 "◊IN."
  ;: CALL - 868: GOTO 1000
4999 REM SET CHAR WIDTH FLAGS
5000 IF A$ = CHR$ (4) THEN D = 1:S
  = 0
5010 IF A$ = CHR$ (14) OR A$ =
  CHR$ (12) THEN D = 0:S = 0
5020 IF A$ = CHR$ (19) THEN D =
  0:S = 1
5030 RETURN
5999 REM PRINT LABELS
6000 FOR X = 20 TO 24: VTAB X:
  HTAB 1: CALL - 868: NEXT : VTAB
  21: PRINT : INPUT "HOW◊MANY◊
  LABELS?" ;F: IF F = 0 THEN 6100
6010 PRINT : PRINT CHR$ (4);
  "PR#1"
6020 PRINT CHR$ (27) CHR$ (64);
6030 PRINT CHR$ (27) CHR$ (48 + 2
  * (LI = 6));
6040 FOR Z = 1 TO F: FOR L = 1 TO
  LI
6050 FOR X = 1 TO LEN (Q$(L)):A$ =
  MID$ (Q$(L),X,1)
6060 IF A$ < CHR$ (32) THEN GOSUB
  7000
6070 PRINT A$;: NEXT : PRINT :
  NEXT : NEXT Z: PRINT CHR$ (4)
  "PR#0"
6099 REM END ROUTINE
6100 PRINT "MORE◊LABELS?◊(Y/N)◊"
  ;: GET A$: IF A$ = "N" OR A$ =
  CHR$ (110) THEN END
6110 PRINT "START◊FRESH?◊(Y/N)◊"
  ;: GET A$: IF A$ = "N" OR A$ =
  CHR$ (110) THEN FOR X = 21 TO
  24: VTAB X: HTAB 1: CALL - 868:
  NEXT : GOTO 1000
6120 RUN 230
6999 REM CTRL-CHAR TURN INTO
  PRINTER CODES
7000 IF A$ = CHR$ (2) THEN A$ =
  CHR$ (27) + CHR$ (71)
7010 IF A$ = CHR$ (4) THEN A$ =
  CHR$ (27) + CHR$ (14)
7020 IF A$ = CHR$ (9) THEN A$ =
  CHR$ (27) + CHR$ (52)
7030 IF A$ = CHR$ (12) THEN A$ =
  CHR$ (27) + CHR$ (120) + "1" +
  CHR$ (27) + CHR$ (107) + "1"
7040 IF A$ = CHR$ (14) THEN A$ =
  CHR$ (27) + CHR$ (33) + CHR$
  (0) + CHR$ (27) + CHR$ (120) +
  "0"
7050 IF A$ = CHR$ (19) THEN A$ =
  CHR$ (27) + CHR$ (15)
7060 IF A$ = CHR$ (31) THEN A$ =
  CHR$ (27) + CHR$ (45) + "1"
7070 RETURN
```

Checksums

```
 10-$BADD  1030-$B480  6000-$7F2D
 20-$9B13  1040-$10E3  6010-$2C6B
 30-$4D3B  1050-$8193  6020-$C03E
 40-$AD92  1060-$26AD  6030-$5381
 50-$C899  1999-$ABC3  6040-$3725
 60-$FF65  2000-$81E6  6050-$CA48
100-$621E  2010-$2C44  6060-$DB08
110-$65FE  2020-$1347  6070-$52BF
120-$EE8B  2030-$CE48  6099-$0965
130-$55CB  2040-$78E0  6100-$C961
140-$2084  2999-$DCBF  6110-$E556
200-$5750  3000-$5EC8  6120-$2715
210-$7A9E  3010-$2283  6999-$ADD6
220-$CFDD  3020-$7074  7000-$DB1F
230-$07CE  3030-$0AA7  7010-$628D
240-$54CD  4999-$EBEB  7020-$FE59
```

250-$EFDB	5000-$7CD7	7030-$D968
999-$E782	5010-$F29A	7040-$FB80
1000-$800B	5020-$CA8B	7050-$78C9
1010-$A828	5030-$F1A0	7060-$22FB
1020-$678D	5999-$8252	7070-$F56C

Note to Personal Newsletter Users

Trying to eliminate clutter on my desk by combining disks, I came across something interesting on the back of my Personal Newsletter Clip-Art Vol II disk. I was in Copy II+ and happened to have the show deleted files option going when I spotted ten deleted font files. Hmmm. I immediately tried Undelete from the menu — no luck. However, with a little persistence I was able to recover all of them.

I noticed after mapping the disk that there were exactly enough blocks between existing files to allow for the deleted ones. It would seem they hadn't been over written yet, I wonder why they were not recoverable.

I became very curious so I went to my reference library and pulled out Beneath Apple ProDOS (BAP) and started to read. Although I am intimately familiar with DOS and file structure of 3.3 disks, thanks to Beneath Apple DOS, I don't really understand ProDOS and its strange blocks from sectors arrangement. I did know that one Block is equal to two sectors. When a block is read in, it actually loads two sectors.

Knowing what I needed to look into, I turned to chapter four on Volumes, Directories and Files. There I learned that ProDOS is very similar to DOS 3.3 in the way disk space and files are kept track of. Most importantly I learned that a $00 is put over the first byte of the file entry upon deletion. This byte tells file type and length of the filename. I know the fonts are binary, so no problem.

I used the Zap program from BAP to edit the disk blocks with the file names. If you don't have a block editor don't worry, I calculated sectors for you to edit. I then tried to copy the files from the disk and was met unanimously with errors by all my file copiers. DOS 3.3 never cared, but ProDOS won't let you copy files unless the blocks they occupy are marked as used in the VTOC. I didn't want to calculate which blocks were in use so I marked the entire disk used, what a pain.

In a nutshell, copy the Clip-Art disk side B, edit the disk, then copy the files to a data disk. Enjoy the fonts that there was room for but they just didn't want you to have.

Changeall of the following:

COLOSSAL.FNT (trk 0, sec B)

Blk	Byte	From	To
2	A0	00	2C

CYRILLIC.FNT

Blk	Byte	From	To
2	EE	00	2C

ESPERANTO.FNT (trk 0, sec A)

Blk	Byte	From	To
2	115	00	2D

FLOW.FNT

Blk	Byte	From	To
2	13C	00	28

GRAPHIC.FNT

Blk	Byte	From	To
2	163	00	2B

GREEK.FNT

Blk	Byte	From	To
2	18A	00	29

KATAKANA.FNT

Blk	Byte	From	To
2	1B1	00	2C

PINOCCHIO.FNT

Blk	Byte	From	To
2	1D8	00	2D

PUDGY.FNT (trk 0, sec 9)

Blk	Byte	From	To
3	04	00	29

STOP.FNT

Blk	Byte	From	To
3	52	00	28

Block 6 Usage map (trk 0, sec 3)

Blk	Byte	From	To
6	00-FF	??	all 00

This will fix FLOW (trk 8,sec 7)

Blk	Byte	From	To
44	00	00 00 00	43 45 46

(Trk 8,Sec 6)

Blk	Byte	From	To
	100	43 45 46	00 00 00

Gary Wills — Canada

Boy! Am I looking forward to a Monthly Computist! Also, I have a few "Christmas Wishes".

? Has anyone successfully put Omega (//e) onto a 3.5" disk. I have met with limited success. I am able to basically run Omega but must use the 5.25" disks when the Resource & Samples directories are needed. I had to rename all of the prefixes for the 3.5" and put all of the files into subdirectories. Unfortunately, when the program searches drives for the appropriate disk, it doesn't access the subdirectories and, therefore, can't find the correct file. Even putting ALL of the files into ONE subdirectory doesn't seem to do the trick.

? "Where in Time is Carmen Sandiego?" - Is there any info on how to get more time (ie. allotted time + 10 hrs) to solve the crimes. My students find it frustrating and are not getting into the program as they did with "Carmen World".

? Is there any interest on creating a "Robotics Interface", if it were inexpensive/feasible?

? I have been working on "Test Drive" (//e) for the last 3 years! No success. Please add it to the "Most Wanted" list.

? Is there any info on "Chessmaster 2100" (//e)? I have a friend who purchased it last year & is afraid of a bomb.

? I work on a Laser EX (ROM 5.0) and have found that The New Print Shop (TNPS) is not compatible with it. Friends who own the Laser EX/2 do not seem to experience these problems. Since I have 2 EX's and the problems are identical, it seems to indicate a hardware/software glitch. When printing out, I am prompted to "Press Return" after every 3 lines printed. Just imagine printing a banner! Although I haven't delved too deeply into TNPS, I also noticed that the Convert utility has a scrambled screen also.

I would appreciate any and all help!

Kim Griffith — CA

Printing Maps for Under Fire

If you have ever played Under Fire, you have probably noticed that the on-screen maps don't always closely match what the terrain truly is. Sometimes what is shown on-screen doesn't even exist if you check the terrain. The map printing routine included on the Under Fire Map Maker disk only duplicates what the screen shows, not what the terrain actually is. The following program will read the map data from disk and print a block type map that will show what exists on the map as the computer sees it. The program was written for an ImageWriter II with a color ribbon, compatibility with other printers is unknown.

When the map is printed, hedges will be green; stone walls, red; stone buildings, dark purple; wood buildings, dark orange; water, blue; rough, textured purple; level 1, white; level 2, yellow; level 4, light orange; roads, black; trees have a green border with a green dot in the center which allows the elevation to be seen inside the green border. Buildings, roads, hedges, walls, etc., don't indicate elevation because too many different combinations of colors would be necessary which would make the map too difficult to read, thus confusing. The map will give you sufficient information so you can check more precisely during game play or with the Map Maker program.

Entering the program

First, get into the monitor and enter the machine language code.

MAPSRC.OBJ0

CALL-151

```
7000:4C 5A 74 1B 4B 30 1B 70  $916E
7008:1B 3E 1B 54 31 36 1B 47  $A377
7010:30 37 33 38 30 31 32 33  $BA34
7018:34 35 36 1E 05 06 07 75  $26DE
7020:76 77 90 95 96 97 A0 A5  $6AC2
7028:A6 A7 B0 B5 B6 B7 D0 D5  $461E
7030:D6 D7 E0 E5 E6 E7 F0 F5  $D21A
7038:F6 F7 08 01 71 91 A1 B1  $4F45
7040:D1 E1 F1 08 02 72 92 A2  $8671
7048:B2 D2 E2 F2 08 03 73 93  $09CA
7050:A3 B3 D3 E3 F3 08 04 74  $9EBD
7058:94 A4 B4 D4 E4 F4 08 10  $B44B
7060:11 12 13 14 15 16 17 08  $AC77
7068:20 21 22 23 24 25 26 27  $6C47
7070:08 30 31 32 33 34 35 36  $B336
7078:37 08 40 41 42 43 44 45  $8AE2
7080:46 47 08 50 51 52 53 54  $2F02
7088:55 56 57 08 60 61 62 63  $077E
7090:64 65 66 67 08 80 81 82  $304F
7098:83 84 85 86 87 08 C0 C1  $714B
70A0:C2 C3 C4 C5 C6 C7 FF FF  $AE1B
70A8:FF FF FF FF FF FF FF FF  $CE5B
70B0:FF FF FF FF 7F 7F 7F 7F  $2E9B
70B8:7F 7F 7F 7F 7F 7F 7F 7F  $4EDB
70C0:7F 7F 2A 2A 55 55 2A 2A  $6B26
70C8:55 55 2A 2A 55 55 2A 2A  $5B4E
70D0:7F 7F 41 41 41 41 49 49  $7065
70D8:41 41 41 41 7F 7F 00 00  $90E7
70E0:3E 3E 3E 3E 36 36 3E 3E  $0C3B
70E8:3E 3E 00 00 00 00 2A 2A  $88DB
70F0:14 14 2A 2A 14 14 2A 2A  $9457
70F8:00 00 A9 86 85 FF A9 8A  $97DD
7100:85 FB A9 00 85 FE 85 FA  $0A00
7108:8D C0 89 4C 1B 71 BE 00  $A1FB
7110:C1 8E 19 71 A2 C1 A0 10  $9AB2
7118:4C 00 C1 A0 0D 20 0E 71  $0284
7120:A0 00 B9 03 70 20 2E 71  $9972
7128:C8 C0 0B D0 F5 60 20 3A  $5ADD
7130:71 A0 0F 20 0E 71 20 44  $CD07
7138:71 60 8D C1 89 8E C2 89  $FC1F
7140:8C C3 89 60 AD C1 89 AE  $1658
7148:C2 89 AC C3 89 60 A9 80  $CDD3
7150:85 FD 20 7D 71 60 A0 0E  $9952
7158:A9 FF 91 FE C8 91 FE 98  $1240
7160:18 69 0F A8 C9 0E D0 F0  $2E3F
7168:E6 FF A5 FF C9 89 D0 E8  $FEBF
7170:A9 86 85 FF 60 A9 00 85  $1103
7178:FD 20 7D 71 60 A0 00 A5  $9309
7180:FD 91 FE C8 D0 FB E6 FF  $CE1A
7188:A5 FF C9 89 D0 EF A9 86  $6344
7190:85 FF 60 A0 00 B9 0E 70  $3ED0
7198:20 2E 71 C8 C0 06 D0 F5  $D903
71A0:A5 FD 20 2E 71 20 2E 71  $9E86
71A8:A0 00 B1 FE 20 2E 71 C8  $B06E
71B0:D0 F8 E6 FF A5 FF C9 88  $61D6
71B8:D0 F0 B1 FE 20 2E 71 C8  $C746
71C0:C0 E0 D0 F6 A9 0D 20 2E  $6E2F
71C8:71 A9 86 85 FF 60 A2 2D  $4645
71D0:AC C0 89 B1 FA 9D D0 89  $7E90
71D8:CA A5 FA 18 69 40 85 FA  $519B
71E0:D0 F1 E6 FB A5 FB C9 96  $4BFA
71E8:D0 E9 C8 8C C0 89 A9 8A  $A380
71F0:85 FB 60 A9 1B 20 2E 71  $B699
71F8:A9 4B 20 2E 71 8A 20 2E  $4BB7
7200:71 60 20 75 71 AE 15 70  $0A5F
7208:20 F3 71 A2 00 BD D0 89  $C734
7210:C9 62 D0 06 20 34 74 4C  $C38B
7218:2B 72 AC 43 70 D9 43 70  $9DF7
7220:D0 06 20 FB 73 4C 2B 72  $FE24
7228:88 D0 F2 E8 E0 2E D0 DD  $46CF
7230:A9 00 85 FD 20 93 71 20  $BC9F
7238:75 71 AE 16 70 20 F3 71  $90E6
7240:A2 00 BD D0 89 C9 63 D0  $437F
7248:06 20 47 74 4C 71 72 AC  $6FD0
7250:4C 70 D9 4C 70 D0 06 20  $AC65
7258:E8 73 4C 71 72 88 D0 F2  $FD57
7260:AC 94 70 D9 94 70 D0 06  $6232
7268:20 FB 73 4C 71 72 88 D0  $583C
7270:F2 E8 E0 2E D0 CC A9 00  $B184
7278:85 FD 20 93 71 20 75 71  $1424
7280:AE 17 70 20 F3 71 A2 00  $2F28
7288:BD D0 89 C9 61 D0 06 20  $4D6B
7290:47 74 4C B7 72 AC 3A 70  $EF97
7298:D9 3A 70 D0 06 20 E8 73  $F3DA
72A0:4C B7 72 88 D0 F2 AC 5E  $256B
72A8:70 D9 5E 70 D0 06 20 FB  $C8E1
72B0:73 4C B7 72 88 D0 F2 E8  $DB46
72B8:E0 2E D0 CC A9 00 85 FD  $4691
72C0:20 93 71 20 75 71 AE 18  $F1E2
72C8:70 20 F3 71 A2 00 BD D0  $0618
72D0:89 C9 64 D0 06 20 47 74  $1796
72D8:4C 0E 73 AC 55 70 D9 55  $2716
72E0:70 D0 06 20 E8 73 4C 0E  $659C
72E8:73 88 D0 F2 AC 67 70 D9  $D485
72F0:67 70 D0 06 20 FB 73 4C  $F74D
72F8:0E 73 88 D0 F2 AC 70 70  $60D8
7300:D9 70 70 D0 06 20 FB 73  $6303
7308:4C 0E 73 88 D0 F2 E8 E0  $52D2
7310:2E D0 BB A9 00 85 FD 20  $36FC
7318:93 71 20 75 71 AE 19 70  $09C2
7320:20 F3 71 A2 00 BD D0 89  $B4B9
7328:AC 8B 70 D9 8B 70 D0 06  $E15C
7330:20 21 74 4C 4A 73 88 D0  $CED3
7338:F2 AC 9D 70 D9 9D 70 D0  $920D
7340:06 20 FB 73 4C 4A 73 88  $68B1
7348:D0 F2 E8 E0 2E D0 D6 A9  $C162
7350:00 85 FD 20 93 71 20 75  $C976
7358:71 AE 1A 70 20 F3 71 A2  $A97D
7360:00 BD D0 89 AC 1B 70 D9  $5F2A
7368:1B 70 D0 06 20 E8 73 4C  $9F7D
7370:97 73 88 D0 F2 AC 79 70  $92B0
7378:D9 79 70 D0 06 20 FB 73  $608D
7380:4C 97 73 88 D0 F2 AC 82  $3867
7388:70 D9 82 70 D0 06 20 FB  $4BB1
7390:73 4C 97 73 88 D0 F2 E8  $ADBE
7398:E0 2E D0 C5 A9 00 85 FD  $4DB5
73A0:20 93 71 20 75 71 AE 14  $8600
73A8:70 20 F3 71 20 4E 71 20  $A2DD
73B0:56 71 A2 00 BD D0 89 29  $7875
73B8:0F C9 08 90 03 20 0E 74  $38BB
73C0:E8 E0 2E D0 EF A9 FF 85  $093B
73C8:FD 20 93 71 60 8A 0A 0A  $DAA3
73D0:0A 0A 85 FE 8A 4A 4A 4A  $E392
73D8:4A 18 65 FF 85 FF 60 A9  $5B14
73E0:00 85 FE A9 86 85 FF 60  $83D9
73E8:20 CD 73 A0 00 B9 C2 70  $A9AE
73F0:91 FE C8 C0 0E D0 F6 20  $C9EE
73F8:DF 73 60 20 CD 73 A0 00  $4C6E
7400:B9 B4 70 91 FE C8 C0 0E  $107E
7408:D0 F6 20 DF 73 60 20 CD  $1B65
7410:73 A0 00 B9 A6 70 91 FE  $D43E
7418:C8 C0 0E D0 F6 20 DF 73  $D2F6
7420:60 20 CD 73 A0 00 B9 D0  $4F0B
7428:70 91 FE C8 C0 0E D0 F6  $F70B
7430:20 DF 73 60 20 CD 73 A0  $B76E
7438:00 B9 DE 70 91 FE C8 C0  $5538
7440:0E D0 F6 20 DF 73 60 20  $EF44
7448:CD 73 A0 00 B9 EC 70 91  $45CD
7450:FE C8 C0 0E D0 F6 20 DF  $0B4A
7458:73 60 20 FA 70 20 75 71  $21AF
7460:20 4E 71 A9 0D 20 2E 71  $0AD2
7468:A9 80 85 FD 20 93 71 A9  $3916
7470:0A 20 2E 71 20 CE 71 20  $AC62
7478:02 72 A9 0A 20 2E 71 AD  $BE3B
7480:C0 89 C9 40 D0 EE 60     $9B1F
```

BSAVE MAPSRC.OBJ0, A$7000, L$487

The Applesoft program will handle the rest of the chores. When prompted for the map name, don't type any suffixes, just the name that has been assigned to the map.

If anyone has any comments or suggestions, I can be reached at:

4916 N. Rio Hondo
Temple City, CA 91780

Under Fire Mapper

FP

```
10 IF PEEK (28672) = 76 THEN 30
20 PRINT CHR$ (4); "BLOAD◊MAPSRC.
   OBJ0"
30 INPUT "ENTER◊NAME◊OF◊MAP◊TO◊
   PRINT◊:" ;N$
40 PRINT CHR$ (4); "BLOAD◊" ;N$;
   ".DAT,A$8A00"
45 CALL 28672
50 INPUT "PRINT◊S)AME◊/◊D)IFFERENT◊
   /◊Q)UIT◊" ;A$
60 IF A$ = "S" THEN 45
70 IF A$ = "D" THEN 30
80 IF A$ < > "Q" THEN 50
90 END
```

SAVE UNDER FIRE MAPPER

Checksums

10-$AE5F	45-$6B78	80-$26C9
20-$62B9	50-$8575	90-$35F5
30-$9E14	60-$8407	
40-$3A61	70-$B3E7	

MAPSRC Source Code

```
*MAPSRC — Prints Under Fire block terrain maps
**** PUT EQUs HERE ****
MAPPTR      EQU $FA         ;Pointer to map data
TMPSAV      EQU $FD         ;Temporary storage
BUFFPTR     EQU $FE         ;Pointer to printer data
COLDAT      EQU $89D0       ;Map column data buffer (L$,2E)
COLCTR      EQU $89C0       ;Map column counter
ASAVE       EQU $89C1       ;Accumulator save location
XSAVE       EQU $89C2       ;X register save location
YSAVE       EQU $89C3       ;Y register save location
**** START HERE ****
            ORG $7000
**** JUMP TO MAIN PROGRAM ****
            JMP  MAINPROG
**** DATA HERE ****
INITDATA    DFB  $1B,$4B,$30          ;Black code
            DFB  $1B,$70              ;144 DPI code
            DFB  $1B,$3E              ;Uni-directional
            DFB  $1B,$54,$31,$36      ;Set line feed to 16/144 inch
GRFXDATA    DFB  $1B,$47              ;Graphics command
            DFB  $30,$37,$33,$38      ;Number of bytes to print
BLACK       DFB  $30                  ;BLACK code
YELLOW      DFB  $31                  ;YELLOW code
MAGENTA     DFB  $32                  ;MAGENTA code
CYAN        DFB  $33                  ;CYAN code
ORANGE      DFB  $34                  ;ORANGE code
GREEN       DFB  $35                  ;GREEN code
PURPLE      DFB  $36                  ;PURPLE code
;Map byte values:
ROUGHDAT    DFB  $1E,$05,$06,$07,$75,$76,$77
            DFB  $90,$95,$96,$97,$A0,$A5,$A6,$A7
            DFB  $B0,$B5,$B6,$B7,$D0,$D5;$D6,$D7
            DFB  $E0,$E5,$E6,$E7,$F0,$F5,$F6,$F7
OPEN0DAT    DFB  $08,$01,$71,$91,$A1,$B1,$D1,$E1,$F1
OPEN2DAT    DFB  $08,$02,$72,$92,$A2,$B2,$D2,$E2,$F2
OPEN3DAT    DFB  $08,$03,$73,$93,$A3,$B3,$D3,$E3,$F3
OPEN4DAT    DFB  $08,$04,$74,$94,$A4,$B4,$D4,$E4,$F4
WATERDAT    DFB  $08,$10,$11,$12,$13,$14,$15,$16,$17
WDBLD1DAT   DFB  $08,$20,$21,$22,$23,$24,$25,$26,$27
WDBLD2DAT   DFB  $08,$30,$31,$32,$33,$34,$35,$36,$37
STBLD1DAT   DFB  $08,$40,$41,$42,$43,$44,$45,$46,$47
STBLD2DAT   DFB  $08,$50,$51,$52,$53,$54,$55,$56,$57
WOODSDAT    DFB  $08,$60,$61,$62,$63,$64,$65,$66,$67
STNWALDAT   DFB  $08,$80,$81,$82,$83,$84,$85,$86,$87
HEDGEDAT    DFB  $08,$C0,$C1,$C2,$C3,$C4,$C5,$C6,$C7
;Graphics byte values for printer:
SQROAD      DFB  %11111111,%11111111,%11111111,%11111111
            DFB  %11111111,%11111111,%11111111,%11111111
            DFB  %11111111,%11111111,%11111111,%11111111
            DFB  %11111111,%11111111
SQSOLID     DFB  %01111111,%01111111,%01111111,%01111111
            DFB  %01111111,%01111111,%01111111,%01111111
            DFB  %01111111,%01111111,%01111111,%01111111
            DFB  %01111111,%01111111
SQDOTTED    DFB  %00101010,%00101010,%01010101,%01010101
            DFB  %00101010,%00101010,%01010101,%01010101
            DFB  %00101010,%00101010,%01010101,%01010101
            DFB  %00101010,%00101010
SQBORDER    DFB  %01111111,%01111111,%01000001,%01000001
            DFB  %01000001,%01000001,%01001001,%01001001
            DFB  %01000001,%01000001,%01000001,%01000001
            DFB  %01111111,%01111111
SQCENTER    DFB  %00000000,%00000000,%00111110,%00111110
            DFB  %00111110,%00111110,%00110110,%00110110
            DFB  %00111110,%00111110,%00111110,%00111110
            DFB  %00000000,%00000000
SQDOTCENT   DFB  %00000000,%00000000,%00101010,%00101010
            DFB  %00010100,%00010100,%00101010,%00101010
            DFB  %00010100,%00010100,%00101010,%00101010
            DFB  %00000000,%00000000
**** SUBROUTINE TO INITIALIZE RAM LOCATIONS ****
START       LDA  #$86
            STA  BUFFPTR+1      ;BUFFPTR points to $8600
            LDA  #$8A
            STA  MAPPTR+1       ;MAPPTR points to $8A00
            LDA  #$00
            STA  BUFFPTR
            STA  MAPPTR
            STA  COLCTR         ;Column counter is zero
;INITIALIZE PRINTER
            JMP  BEGIN
GOCARD      LDX  $C100,Y        ;Get lo byte of entry point
            STX  VECTOR+1       ;Self modifying code
            LDX  #$C1           ;Required; X = Cn
            LDY  #$10           ;Required; Y = n0
VECTOR      JMP  $C100          ;This address gets modified
BEGIN       LDY  #$0D           ;Initialize....
            JSR  GOCARD         ;..card
            LDY  #$00           ;Init counter
DATALOOP    LDA  INITDATA,Y     ;Black, 144 DPI, uni-dir., line feed
            JSR  SENDBYTE       ;Save registers
            INY                 ;Increment counter
            CPY  #$0B           ;Have all bytes been sent?
            BNE  DATALOOP       ;If not, do again
            RTS                 ;Return
**** SUBROUTINE TO SEND BYTE TO PRINTER ****
SENDBYTE    JSR  SAVEREGS       ;Save registers
            LDY  #$0F           ;Necessary
            JSR  GOCARD         ;Send byte to card
            JSR  RESTREGS       ;Restore registers
            RTS
**** SUBROUTINE TO SAVE REGISTERS ****
SAVEREGS    STA  ASAVE          ;Save accumulator
            STX  XSAVE          ;Save X register
            STY  YSAVE          ;Save Y register
            RTS
**** SUBROUTINE TO RESTORE REGISTERS ****
RESTREGS    LDA  ASAVE          ;Restore accumulator
            LDX  XSAVE          ;Restore X register
            LDY  YSAVE          ;Restore Y register
            RTS
**** SUBROUTINE TO PUT NORTH-SOUTH LINE IN BUFFER ****
NSLINE      LDA  #%10000000     ;Value for printer
            STA  TMPSAV         ;Put it where BUFFILL can find it
            JSR  BUFFILL        ;Go fill the buffer
            RTS
**** SUBROUTINE TO PUT EAST-WEST LINES IN BUFFER ****
EWLINES     LDY  #$0E           ;Initialize for offset into buffer
LOOPEW      LDA  #%11111111     ;Value for E-W map lines
            STA  (BUFFPTR),Y    ;Put it in buffer
            INY                 ;Increment offset
            STA  (BUFFPTR),Y    ;Put it in buffer again
            TYA                 ;Increment Y reg 0F times:
            CLC
            ADC  #$0F           ;Add 0F
            TAY
            CMP  #$0E           ;Is Y reg 0E yet?
            BNE  LOOPEW         ;If not, do another
            INC  BUFFPTR+1      ;Increment hi byte of pointer
            LDA  BUFFPTR+1      ;Get hi byte of pointer
            CMP  #$89           ;Is it (page) 89?
            BNE  LOOPEW         ;If not, then less. Do it again
            LDA  #$86           ;Restore pointer in zero page
            STA  BUFFPTR+1
            RTS                 ;Return
**** SUBROUTINE TO ZERO PRINTER BUFFER ****
ZEROBUFF    LDA  #$00           ;Get 00 value
            STA  TMPSAV         ;Store it where BUFFILL can find it
            JSR  BUFFILL        ;Fill the buffer with it
            RTS                 ;Return
**** SUBROUTINE TO FILL BUFFER WITH VALUE IN TMPSAV ****
BUFFILL     LDY  #$00           ;Init counter for index into buffer
            LDA  TMPSAV         ;Value to fill buffer with
LOOP1       STA  (BUFFPTR),Y    ;Put it in buffer
            INY                 ;Increment counter
            BNE  LOOP1          ;Do again until page is full
            INC  BUFFPTR+1      ;Increment MSB of pointer
            LDA  BUFFPTR+1      ;Get MSB of pointer
            CMP  #$89           ;Is it 89?
            BNE  BUFFILL        ;If not, then it's less; do again
            LDA  #$86
            STA  BUFFPTR+1      ;Restore pointer in zero page
            RTS                 ;Return
**** SUBROUTINE TO SEND PRINTER BUFFER TO PRINTER ****
PRINTBUFF   LDY  #$00           ;Initialize counter
;First, set up to print a line of graphics:
GRFXLOOP    LDA  GRFXDATA,Y     ;Get a byte
            JSR  SENDBYTE       ;Send the byte
            INY                 ;Increment counter
            CPY  #$06           ;Have 06 bytes passed?
            BNE  GRFXLOOP       ;If not, get another
;Now send 738 graphics bytes to the printer:
            LDA  TMPSAV         ;Get value for first two bytes
            JSR  SENDBYTE       ;Send it....
            JSR  SENDBYTE       ;..twice
            LDY  #$00           ;Initialize counter to send a page
LOOP2       LDA  (BUFFPTR),Y    ;Get byte from buffer
            JSR  SENDBYTE       ;Send the byte
            INY                 ;Increment counter
            BNE  LOOP2          ;If page not done yet, do again
            INC  BUFFPTR+1      ;Increment page
            LDA  BUFFPTR+1      ;Get page
            CMP  #$88           ;Is it 88?
            BNE  LOOP2          ;If not, do another page
LOOP3       LDA  (BUFFPTR),Y    ;Get byte from buffer
            JSR  SENDBYTE       ;Send the byte
            INY                 ;Increment counter
            CPY  #$E0           ;Have E0 bytes passed?
            BNE  LOOP3          ;If not, get another byte
            LDA  #$0D           ;CARRIAGE RETURN
            JSR  SENDBYTE       ;Send it to printer
            LDA  #$86           ;Restore pointer in zero page
            STA  BUFFPTR+1
            RTS                 ;Return
**** SUBROUTINE TO GET MAP COLUMN DATA & PUT IN COLUMN
                                BUFFER ****
GETMAPCOL   LDX  #$2D           ;For index into COLDAT
            LDY  COLCTR         ;For use as column offset in map data
LOOP4       LDA  (MAPPTR),Y     ;Get map byte
            STA  COLDAT,X       ;Put in buffer (hi to lo)
            DEX                 ;Decrement index into buffer
            LDA  MAPPTR         ;Get lo byte of map pointer
            CLC
            ADC  #$40           ;Incr. map buffer pointer by 1 map row
            STA  MAPPTR         ;Store it at proper location
            BNE  LOOP4          ;If not 0 (4 rows done), do another row
            INC  MAPPTR+1       ;Increment hi byte of map pointer
            LDA  MAPPTR+1       ;Get hi byte of map pointer
            CMP  #$96           ;Is it 96?
            BNE  LOOP4          ;If not, go do another page
            INY                 ;Increment column number
            STY  COLCTR         ;Save it at proper location
            LDA  #$8A
            STA  MAPPTR+1       ;Restore map pointer in zero page
            RTS
**** SUBROUTINE TO SET COLOR TO VALUE IN X REG ****
COLOR       LDA  #$1B           ;Send two necessary bytes:
            JSR  SENDBYTE
            LDA  #$4B
            JSR  SENDBYTE
            TXA                 ;Get color code
            JSR  SENDBYTE       ;Send color byte
            RTS
**** SUBROUTINE TO PRINT ALL COLORS FROM CURRENT COLUMN ****
;YELLOW starts here
PRINTCOL    JSR  ZEROBUFF       ;First, zero the buffer
            LDX  YELLOW         ;Get value for YELLOW
            JSR  COLOR          ;Set printer to YELLOW
            LDX  #$00           ;Initialize index into COLDAT
YELLOWLOOP  LDA  COLDAT,X       ;Get a byte
;Check for map level 2 WOODS (62), CENTER YELLOW
            CMP  #$62           ;Is it level 2 WOODS?
            BNE  CHKOPEN2       ;If not, check level 2 OPEN
            JSR  FILLCENTER     ;It's lvl 2 WOODS, fill center of square
            JMP  NXTYELLOW      ;Go do another square
;Check for map level 2 OPEN, SOLID YELLOW
CHKOPEN2    LDY  OPEN2DAT       ;Get counter for loop
LOOPOPEN2   CMP  OPEN2DAT,Y     ;Compare with byte in table
            BNE  NXTBYTE2       ;If not the same, do another
            JSR  FILLSOLID      ;It's level 2 OPEN, fill solid square
            JMP  NXTYELLOW      ;Go do another square
NXTBYTE2    DEY                 ;Decrement counter
            BNE  LOOPOPEN2      ;If not 0, get another byte from table
NXTYELLOW   INX                 ;Increment index into COLDAT
            CPX  #$2E           ;Have 2E bytes been checked?
            BNE  YELLOWLOOP     ;If not, do another
            LDA  #%00000000     ;Value for first two graphics bytes
            STA  TMPSAV         ;Put it where PRINTBUFF can find it
            JSR  PRINTBUFF      ;Send graphics code, then printer data
;MAGENTA starts here
            JSR  ZEROBUFF       ;First, zero the buffer
            LDX  MAGENTA        ;Set value for MAGENTA
            JSR  COLOR          ;Set printer to MAGENTA
            LDX  #$00           ;Initialize index into COLDAT
MAGENTALOOP LDA  COLDAT,X       ;Get a byte
;Check for map level 3 WOODS (63), CENTER DOTTED MAGENTA
            CMP  #$63           ;Is it level 3 WOODS?
            BNE  CHKOPEN3       ;If not, check level 3 OPEN
            JSR  FILLDOTCNT     ;It's level 3 WOODS, fill dotted center
            JMP  NXTMAGENTA     ;Go do another square
;Check for map level 3 OPEN, SOLID DOTTED MAGENTA
CHKOPEN3    LDY  OPEN3DAT       ;Get counter for loop
LOOPOPEN3   CMP  OPEN3DAT,Y     ;Compare with byte in table
            BNE  NXTBYTE3       ;If not the same, do another
            JSR  FILLDOTTED     ;It's level 3 OPEN, fill dotted square
            JMP  NXTMAGENTA     ;Go do another square
NXTBYTE3    DEY                 ;Decrement counter
            BNE  LOOPOPEN3      ;If not 0, get another byte from table
;Check for STONE WALL ($80.$87)
            LDY  STNWALDAT      ;Get counter for loop
LOOPSW      CMP  STNWALDAT,Y    ;Compare with byte in table
            BNE  NXTBYTESW      ;If not the same, do another
            JSR  FILLSOLID      ;It's STONE WALL, fill solid square
            JMP  NXTMAGENTA     ;Go do another square
NXTBYTESW   DEY                 ;Decrement counter
            BNE  LOOPSW         ;If not 0, get another byte from table
NXTMAGENTA  INX                 ;Increment index into COLDAT
            CPX  #$2E           ;Have 2E bytes passed yet?
            BNE  MAGENTALOOP    ;If not, do another
            LDA  #%00000000     ;Value for first two graphics bytes
            STA  TMPSAV         ;Put it where PRINTBUFF can find it
            JSR  PRINTBUFF      ;Send graphics code, then printer data
;CYAN starts here
            JSR  ZEROBUFF       ;First, zero the buffer
            LDX  CYAN           ;Set value for CYAN
            JSR  COLOR          ;Set printer to CYAN
            LDX  #$00           ;Initialize index into COLDAT
CYANLOOP    LDA  COLDAT,X       ;Get a byte
;Check for map level 0 WOODS (61), CENTER DOTTED CYAN
            CMP  #$61           ;Is it level 0 WOODS?
            BNE  CHKOPEN0       ;If not, check level 0 OPEN
            JSR  FILLDOTCNT     ;It's lvl 0 WOODS, fill center of square
            JMP  NXTCYAN        ;Go do another square
;Check for map level 0 OPEN, SOLID DOTTED CYAN
CHKOPEN0    LDY  OPEN0DAT       ;Get counter for loop
LOOPOPEN0   CMP  OPEN0DAT,Y     ;Compare with byte in table
            BNE  NXTBYTE0       ;If not the same, do another
            JSR  FILLDOTTED     ;It's level 2 OPEN, fill solid square
            JMP  NXTCYAN        ;Go do another square
NXTBYTE0    DEY                 ;Decrement counter
            BNE  LOOPOPEN0      ;If not 0, get another byte from table
;Check for WATER (10.17), SOLID CYAN
            LDY  WATERDAT       ;Get counter for loop
LOOPWATER   CMP  WATERDAT,Y     ;Compare with byte in table
            BNE  NXTWATER       ;If not the same, do another
            JSR  FILLSOLID      ;It's WATER, fill solid square
            JMP  NXTCYAN        ;Go do another square
NXTWATER    DEY                 ;Decrement counter
            BNE  LOOPWATER      ;If not 0, get another byte from table
NXTCYAN     INX                 ;Increment index into COLDAT
            CPX  #$2E           ;Have 2E bytes been checked?
            BNE  CYANLOOP       ;If not, do another
            LDA  #%00000000     ;Value for first two graphics bytes
            STA  TMPSAV         ;Put it where PRINTBUFF can find it
            JSR  PRINTBUFF      ;Send graphics code, then printer data
;ORANGE starts here
            JSR  ZEROBUFF       ;First, zero the buffer
            LDX  ORANGE         ;Set value for ORANGE
            JSR  COLOR          ;Set printer for ORANGE
            LDX  #$00           ;Initialize index into COLDAT
ORANGELOOP  LDA  COLDAT,X       ;Get a byte
;Check for level 4 WOODS (64), CENTER DOTTED ORANGE
            CMP  #$64           ;Is it level 4 WOODS?
            BNE  CHKOPEN4       ;If not, check for level 4 OPEN
```

```
            JSR  FILLDOTCNT   ;It's lvl 4 WOODS, fill center of square
            JMP  NXTORANGE    ;Go check next square
;Check for level 4 OPEN (from LVL4DATA)
CHKOPEN4    LDY  OPEN4DAT     ;Get counter for loop
LOOPOPEN4   CMP  OPEN4DAT,Y   ;Compare with byte in table
            BNE  NXTBYTE4     ;If not the same, do another
            JSR  FILLDOTTED   ;It's level 4 OPEN, fill solid square
            JMP  NXTORANGE    ;Go do another square
NXTBYTE4    DEY               ;Decrement counter
            BNE  LOOPOPEN4    ;If not 0, get another byte from table
;Check for 1 STORY WOODEN BUILDING ($20.$27)
            LDY  WDBLD1DAT    ;Get counter for loop
WOODLOOP1   CMP  WDBLD1DAT,Y  ;Compare with byte in table
            BNE  NXTWBLD1     ;If not the same, do another
            JSR  FILLSOLID    ;It's 2 ST WOOD BLDG, fill solid
                              square
            JMP  NXTORANGE    ;Go do another square
NXTWBLD1    DEY               ;Decrement counter
            BNE  WOODLOOP1    ;If not 0, get another byte from table
;Check for 2 STORY WOODEN BUILDING ($30.$37)
            LDY  WDBLD2DAT    ;Get counter for loop
WOODLOOP2   CMP  WDBLD2DAT,Y  ;Compare with byte in table
            BNE  NXTWBLD2     ;If not the same, do another
            JSR  FILLSOLID    ;It's 1 ST WOOD BLDG, fill solid
                              square
            JMP  NXTORANGE    ;Go do another square
NXTWBLD2    DEY               ;Decrement counter
            BNE  WOODLOOP2    ;If not 0, get another byte from table
NXTORANGE   INX               ;Increment index into COLDAT
            CPX  #$2E         ;Have 2E bytes been checked?
            BNE  ORANGELOOP   ;If not, do another
            LDA  #%00000000   ;Value for first two graphics bytes
            STA  TMPSAV       ;Put it where PRINTBUFF can find it
            JSR  PRINTBUFF    ;Send graphics code, then printer data
;GREEN starts here
            JSR  ZEROBUFF     ;First, zero the buffer
            LDX  GREEN        ;Set value for GREEN
            JSR  COLOR        ;Set printer for GREEN
            LDX  #$00         ;Initialize index into COLDAT
GREENLOOP   LDA  COLDAT,X     ;Get a byte
;Check for WOODS (60.67)
            LDY  WOODSDAT     ;Get counter for loop
LOOPWOODS   CMP  WOODSDAT,Y   ;Compare with byte in table
            BNE  NXTWOODS     ;If not the same, do another
            JSR  FILLBORDER   ;It's WOODS, fill border square
            JMP  NXTGREEN     ;Go do another square
NXTWOODS    DEY               ;Decrement counter
            BNE  LOOPWOODS    ;If not 0, get another byte from table
;Check for HEDGE ($60.$67)
            LDY  HEDGEDAT     ;Get counter for loop
LOOPHEDGE   CMP  HEDGEDAT,Y   ;Compare with byte in table
            BNE  NXTHEDGE     ;If not the same, do another
            JSR  FILLSOLID    ;It's HEDGE, fill solid square
            JMP  NXTGREEN     ;Go do another square
NXTHEDGE    DEY               ;Decrement counter
            BNE  LOOPHEDGE    ;If not 0, get another byte from table
NXTGREEN    INX               ;Increment index into COLDAT
            CPX  #$2E         ;Have 2E bytes been checked?
            BNE  GREENLOOP    ;If not, do another
            LDA  #%00000000   ;Value for first two graphics bytes
            STA  TMPSAV       ;Put it where PRINTBUFF can find it
            JSR  PRINTBUFF    ;Send graphics code, then printer data
;PURPLE starts here
            JSR  ZEROBUFF     ;First, zero the buffer
            LDX  PURPLE       ;Set value for PURPLE
            JSR  COLOR        ;Set printer to PURPLE
            LDX  #$00         ;Initialize index into COLDAT
PURPLELOOP  LDA  COLDAT,X     ;Get a byte
;Check for ROUGH, DOTTED PURPLE
            LDY  ROUGHDAT     ;Get counter for loop
LOOP14      CMP  ROUGHDAT,Y   ;Compare with byte in table
            BNE  NXTROUGH     ;If not the same, do another
            JSR  FILLDOTTED   ;It's ROUGH, fill dotted square
            JMP  NXTPURPLE    ;Go do another square
NXTROUGH    DEY               ;Decrement counter
            BNE  LOOP14       ;If not 0, get another byte from table
;Check for 1 STORY STONE BUILDING ($40.$47)
            LDY  STBLD1DAT    ;Get counter for loop
LOOPSBLD1   CMP  STBLD1DAT,Y  ;Compare with byte in table
            BNE  NXTSBLD1     ;If not the same, do another
            JSR  FILLSOLID    ;It's 1 S STN BLDG, fill solid square
            JMP  NXTPURPLE    ;Go do another square
NXTSBLD1    DEY               ;Decrement counter
            BNE  LOOPSBLD1    ;If not 0, get another byte from table
;Check for 2 STORY STONE BUILDING ($50.$57)
            LDY  STBLD2DAT    ;Get counter for loop
LOOPSBLD2   CMP  STBLD2DAT,Y  ;Compare with byte in table
            BNE  NXTSBLD2     ;If not the same, do another
            JSR  FILLSOLID    ;It's 2 S STN BLDG, fill solid square
            JMP  NXTPURPLE    ;Go do another square
NXTSBLD2    DEY               ;Decrement counter
            BNE  LOOPSBLD2    ;If not 0, get another byte from table
NXTPURPLE   INX               ;Increment index into COLDAT
            CPX  #$2E         ;Have 2E bytes been checked?
            BNE  PURPLELOOP   ;If not, do another
            LDA  #%00000000   ;Value for first two graphics bytes
            STA  TMPSAV       ;Put it where PRINTBUFF can find it
            JSR  PRINTBUFF    ;Send graphics code, then printer data
;BLACK starts here
            JSR  ZEROBUFF     ;First, zero the buffer
            LDX  BLACK        ;Get value for BLACK
            JSR  COLOR        ;Set printer to BLACK
            JSR  NSLINE       ;Put north-south line in buffer
            JSR  EWLINES      ;Put east-west lines in buffer
            LDX  #$00         ;Initialize index into COLDAT
BLACKLOOP   LDA  COLDAT,X     ;Get a byte
;Check for ROAD (lo nibble x8 or greater), SOLID BLACK
            AND  #$0F         ;Mask off hi nibble
            CMP  #$08
            BCC  NXTBLACK     ;If less than 08, check next terrain
            JSR  FILLROAD     ;It's road, fill solid square
NXTBLACK    INX               ;Increment index into COLDAT
            CPX  #$2E         ;Have 2E bytes been checked?
            BNE  BLACKLOOP    ;If not, do another
            LDA  #%11111111   ;Value for first two graphics bytes
            STA  TMPSAV       ;Put it where PRINTBUFF can find it
            JSR  PRINTBUFF    ;Send graphics code, then printer data
            RTS               ;Return
**** SUBROUTINE TO GET PRINTER BUFFER POINTER ****
GETBUFFPTR  TXA               ;Get offset into column buffer
            ASL               ;Multiply by 10:
            ASL
            ASL
            ASL
            STA  BUFFPTR      ;Put in lo byte of pointer
            TXA               ;Get the offset again
            LSR               ;Divide by 10:
            LSR
            LSR
            LSR
            CLC
            ADC  BUFFPTR+1    ;Add page offset to hi byte of pointer
            STA  BUFFPTR+1    ;Store it in hi byte of pointer
            RTS
**** SUBROUTINE TO RESTORE PRINTER BUFFER POINTER ****
RESTPTR     LDA  #$00
            STA  BUFFPTR      ;Restore lo byte
            LDA  #$86
            STA  BUFFPTR+1    ;Restore hi byte
            RTS
**** SUBROUTINE TO FILL DOTTED SQUARE ****
FILLDOTTED  JSR  GETBUFFPTR   ;Get buffer pointer
            LDY  #$00         ;Initialize the pointer offset
LOOPDOTTED  LDA  SQDOTTED,Y   ;Get a byte from table
            STA  (BUFFPTR),Y  ;Put it in buffer
            INY               ;Increment offset from pointer
            CPY  #$0E         ;Have 0E (14) bytes passed yet?
            BNE  LOOPDOTTED   ;If not, go do another
            JSR  RESTPTR      ;Restore pointer
            RTS               ;Return
**** SUBROUTINE TO FILL SOLID SQUARE ****
FILLSOLID   JSR  GETBUFFPTR   ;Get buffer pointer
            LDY  #$00         ;Initialize the pointer offset
LOOPSOLID   LDA  SQSOLID,Y    ;Get a byte from table
            STA  (BUFFPTR),Y  ;Store it in printer buffer
            INY               ;Increment offset
            CPY  #$0E         ;Have 0E (14) bytes passed?
            BNE  LOOPSOLID    ;If not, do another
            JSR  RESTPTR      ;Restore buffer pointer
            RTS               ;Return
**** SUBROUTINE TO FILL ROAD SQUARE ****
FILLROAD    JSR  GETBUFFPTR
            LDY  #$00
LOOPROAD    LDA  SQROAD,Y
            STA  (BUFFPTR),Y
            INY
            CPY  #$0E
            BNE  LOOPROAD
            JSR  RESTPTR
            RTS
**** SUBROUTINE TO FILL BORDER OF SQUARE ****
FILLBORDER  JSR  GETBUFFPTR   ;Get buffer pointer
            LDY  #$00         ;Initialize the pointer offset
LOOPBORDER  LDA  SQBORDER,Y   ;Get a byte from table
            STA  (BUFFPTR),Y  ;Store it in printer buffer
            INY               ;Increment offset
            CPY  #$0E         ;Have 0E (14) bytes passed yet?
            BNE  LOOPBORDER   ;If not, do another
            JSR  RESTPTR      ;Restore buffer pointer
            RTS               ;Return
**** SUBROUTINE TO FILL CENTER OF SQUARE ****
FILLCENTER  JSR  GETBUFFPTR   ;Get buffer pointer
            LDY  #$00         ;Initialize pointer offset
LOOPCENTER  LDA  SQCENTER,Y   ;Get a byte from table
            STA  (BUFFPTR),Y  ;Store it in printer buffer
            INY               ;Increment offset
            CPY  #$0E         ;Have 14 bytes passed?
            BNE  LOOPCENTER   ;If not, go do another
            JSR  RESTPTR      ;Restore buffer pointer
            RTS               ;Return
**** SUBROUTNE TO FILL DOTTED CENTER OF SQUARE ****
FILLDOTCNT  JSR  GETBUFFPTR   ;Get buffer pointer
            LDY  #$00         ;Initialize pointer offset
LOOPDOTCNT  LDA  SQDOTCENT,Y  ;Get a byte from table
            STA  (BUFFPTR),Y  ;Put it in buffer
            INY               ;Increment offset
            CPY  #$0E         ;Have 14 bytes passed?
            BNE  LOOPDOTCNT   ;If not, go do another
            JSR  RESTPTR      ;Restore buffer pointer
            RTS               ;Return
**** MAIN PROGRAM STARTS HERE ****
MAINPROG    JSR  START        ;Initialize pointers, printer
            JSR  ZEROBUFF     ;Zero the printer buffer
            JSR  NSLINE       ;Put north-south line in buffer
            LDA  #$0D         ;CARRIAGE RETURN
            JSR  SENDBYTE     ;Send it to printer
            LDA  #%10000000   ;Value for first two graphics bytes
            STA  TMPSAV       ;Put it where PRINTBUFF can find it
            JSR  PRINTBUFF    ;Send graphics code, then printer data
            LDA  #$0A         ;LINE FEED
            JSR  SENDBYTE     ;Send it to printer
MAINLOOP    JSR  GETMAPCOL    ;Get map column of data, store at
                              COLDAT
            JSR  PRINTCOL     ;Print all colors for the column
            LDA  #$0A         ;LINE FEED
            JSR  SENDBYTE     ;Send it to printer
            LDA  COLCTR       ;Get number of column just printed
            CMP  #$40         ;Is it 3F (64th column)?
            BNE  MAINLOOP     ;If not, do another
            RTS               ;ALL DONE!
```

Suzanne R. Harvey — AZ

I have been playing Wizardry V: The Heart of the Maelstrom for several months but I have now reached an impasse. I would appreciate any help that your readers could give me.

I think I missed several things on the way to Level 7. Is there any significance to the runes which are found on each level?

Level 3

1) A friend told me which buttons to press on the Paradimensional Etherealizer. If you try to discover the right buttons by trial and error, it could take a long time. Is there any clue in the game that tells you which are the right buttons?

2) Is there any significance to the rooms hung with colored curtains?

Level 5

1) When my party entered the Playhouse Mystery Theatre, they fell into a pit at a four-way intersection. There are four rooms here, all the same, and I could not find any way out. What is the purpose of these rooms and is there a way out?

2) My party could not get through the doors on the sides of the Mad Room; whenever they got close, they were teleported away. Is there a way to stop the teleportations and get through the doors?

Level 6

1) Where is Evil Eyes? My party was unable to find him even after three hours of real time searching.

Level 7

1) I have found the four staffs and three of the four cards. Where is the Ace of Clubs?

2) I have discovered the names of two of the Divine Aspects: TIME and NATURE. A friend told me the third Aspect is KINGDOM. Where do you discover this?

3) I am presently at the colored flames at the center of Level 7. I haven't been able to discover what to do here. Can anyone help me?

Advanced Playing Technique for...

Wizardry V

Sir Tech

In playing Wizardry V, I ran into several frustrating circumstances which included the untimely demise of several members of my party. It seemed that a little disk snooping was in order to remedy the situation. I thought the information I discovered would be of interest to some of your readers.

The Wizardry V characters are stored on Disk A, track 4, beginning at sector 7 and going down. Each character takes up about three-quarters of a sector (246 bytes) and then the next character begins. This means a particular piece of information for two different characters will not appear in the same byte relative to the start of a sector but it will appear in the same byte relative to the start of the character's record. Each character's record starts one byte before his name. The locations listed below are relative to this first byte.

Character data

dec	hex	contents/meaning
1	01	Number of bytes in character's name
2-31	02-1F	Character's name
33	21	Race
35	23	Class
37	25	Alignment
39	27	Sex
41	29	Status
43-44	2B-2C	Age
45-48	2D-30	Attributes
53-57	35-38	Gold
59	3B	Number of items carried
61-91	3D-5B	Items carried and their status
93-97	5D-60	Experience points
99-100	63-64	Current level
101-102	65-66	Maximum level
103-104	67-68	Current hit points
105-106	69-6A	Maximum hit points
107-110	6B-6E	Mage spells
111-114	6F-72	Priest spells
115-121	73-79	Mage spell points
123-129	7B-81	Priest spell points
133-134	85-86	Armor class

163-164	A3-A4	Marks
193	C1	Swim counter

Some of this information is stored in a straightforward manner and some is stored very strangely indeed.

Race	Class	Alignment	Sex	Status
01 = Human	00 = fighter	01 = good	01 = male	00 = OK
02 = Elf	01 = mage	02 = neutral	02 = female	01 = OK
03 = Dwarf	03 = priest	03 = evil		02 = slept
04 = Gnome	04 = bishop			03 = paralyzed
05 = Hobbit	05 = samurai			04 = petrified
	06 = lord			05 = dead
	07 = ninja			06 = ashes
				07 = lost

Age

Age is stored in a two-byte lo/hi format. Divide the stored number (in hex) by $34 to find the age. Be aware that there is a range of values which all give the same age when divided by $34. Therefore the stored values for the ages of two characters may be different, even though their displayed ages are the same.

Attributes

The six attributes are stored in four bytes.

To raise this attribute by 1	add	to byte	(hex)	
STRENGTH	$01	45	($2D)	
IQ	$20	45	($2D)	
PIETY	$04	46	($2E)	
VITALITY	$01	47	($2F)	
AGILITY	$20	47	($2F)	
LUCK	$04	48	($30)	

To have all attributes at 18:

Byte location	45 ($2D)	46 ($2E)	47 ($2F)	48 ($30)
Value	52	4A	52	4A

Mage spells

Mage Spells are stored in four bytes.

107 ($6B)	108 ($6C)	109 ($6D)	110 ($6E)	meaning
FE	3F	00	00	All spells L1 - L3
FF	FF	7F	00	All spells L1 - L5
FF	FF	FF	FF	All spells L1 - L7

Priest spells

Priest Spells are stored in four bytes.

111 ($6F)	112 ($70)	113 ($71)	114 ($72)	meaning
FF	FF	FF	0E	All spells L1 - L6 (except Liktofeit)
FF	FF	FF	0F	All spells L1 - L6
FF	FF	FF	FF	All Spells L1 - L7

Armor class

Armor Class is stored in two bytes. If the armor class is positive, Byte 134 ($86) = 00 and Byte 135 ($87) holds the armor class in hex. If the armor class is negative, Byte 134 ($86) = FF and Byte 135 (87) = FF - armor class +1.

Armor class	Byte 134 ($86)	Byte 135 ($87)
3	03	00
2	02	00
1	01	00
0	00	00
-1	FF	FF
-2	FE	FF
-3	FD	FF

Gold

Gold is stored in a modified "256 counter". The first byte counts until it reaches 255 ($FF). When another gold piece is added, this byte goes to zero and the byte to its right goes to one. Then the first byte starts counting again. This continues up to 9999 gp which is stored as 0F 27. 10 000 gp is stored as 00 00 01. As more gold is added, one number is added to the third byte every time the first two bytes reach 0F 27. The third byte goes to FF.

```
0F 27 FF    = 2 559 999 gp
00 00 00 01 = 2 560 000 gp
```

Experience

Experience points are stored in a modified "256 counter" like the gold.

Marks

Marks are stored in a modified "256 counter" like gold and experience points.

Swim Counter

The swim counter counts how many times your character wades in a pool. Your swim stat goes up one for every nine wades.

That's what I've found out so far. I hope it is useful to some of your readers.

Items

Each item carried (max 8) is stored in two bytes separated by $00. The first byte is the item code and the second is the item status.

Item status codes

00 = unidentified
02 = identified
03 = equipped
07 = cursed
0B = armor equipped

Table of item codes

Code	Item	Code	Item	Code	Item
01	Torch	31	Battle axe +2	61	Potion of Latumofis
02	Lantern	32	Axe of Death	62	Potion of Dialko
03	Rubber Duck	33	Sacred Basher	63	Potion of Wounding
04	Dagger	34	Faust Halberd	64	Potion of Madi
05	Staff	35	Silver Hammer	65	King of Diamonds
06	Short Sword	36	Mage's Yew Bow	66	Queen of Hearts
07	Long Sword	37	Hv. Crossbow	67	Jack of Spades
08	Mace	38	Leather +2	68	Ace of Clubs
09	Battle Axe	39	Chain Mail +2	69	Munke Wand
0A	Pike	3A	Scale Mail +2	6A	Lightning Rod
0B	War Hammer	3B	Plate Mail +2	6B	Lark in a Cage
0C	Holy Basher	3C	Scarlet Robes	6C	Staff of Water
0D	Long Bow	3D	Emerald robes	6D	Staff of Fire
0E	Thieves' Bow	3E	Heater +2	6E	Staff of Air
0F	Robes	3F	Bacinet	6F	Staff of Earth
10	Leather Armor	40	Cone of Fire	70	Potion of Demon Out
11	Chain Mail	41	Silver Gloves	71	Gold Medallion
12	Scale Mail	42	Bracers +1	72	Ice Key
13	Plate Mail	43	Long Sword +3	73	Ticket Stubs
14	Target Shield	44	Plate Mail +3	74	Tickets
15	Heater Shield	45	Shield Pro Magi	75	Skeleton Key
16	Leather Sallet	46	Jeweled Armet	76	Pocket Watch
17	Leather Gloves	47	Wizard's Cap	77	Battery
18	Short Sword +1	48	Gloves of Myrdall	78	Petrified Demon
19	Long Sword +1	49	Cloak of Capricorn	79	Gold Key
1A	Black Blade	4A	Sylvan Bow	7A	Blue Candle
1B	Katana	4B	Muramasa Katana	7B	Jeweled Scepter
1C	Battle Axe +1	4C	Odin Sword	7C	Potion of Spirit Away
1D	Morningstar	4D	Gold Plate +5	7D	Hacksaw
1E	Runed Flail	4E	Ring of Froze	7E	Bottle of Rum
1F	Halberd	4F	Ring of Skulls	7F	Silver Key
20	Lt. Crossbow	50	Ring of Madi	80	Bag of Tokens
21	Leather +1	51	Ring of Jade	81	Brass Key
22	Chain Mail +1	52	Ring of Solitude	82	Orb of Llygamyn
23	Scale Mail +1	53	Ankh of Wonder	83	Heart of Abriel
24	Plate Mail +1	54	Ankh of Power	84	Holy Talisman
25	Silver Mail	55	Ankh of Life	85	Amulet of Rainbows
26	Target +1	56	Ankh of Intellect	86	Amulet of Screens
27	Heater +1	57	Ankh of Sanctity	87	Amulet of Flames
28	Crested Shield	58	Ankh of Youth		
29	Brass Sallet	59	Staff of Summoning		
2A	Iron Gloves	5A	Staff of Death		
2B	Bracers	5B	Scroll of Katino		
2C	Long Sword +2	5C	Scroll of Stoning		
2D	Robin Sword	5D	Scroll of Fire		
2E	Sword of Fire	5E	Scroll of Conjuring		
2F	Master Katana	5F	Potion of Dios		

Robin Locksley — MO

Playing Tip for...

Dungeon Master

FTL

The following information provided in the charts will help those who are starting this game and find it a real challenge. Good Luck.

Dungeon Master Data

Bolt Blade	Sword	Lightning (limited)
Calista	Crowm	?
Crown of Nera	Crown	Wisdom plus 10 points
Delta	Sword	Mana plus 1 point
Dragon Spit	Sceptre	Mana plus 7 points
Eye of Time	Ring	Freeze Life (limited)
Firestaff	Staff	Fireshield
Flame Bane	Vest	Anti-fire plus 12 points
Flamitt	Glove	Fireballs
Fury	Sword	Fireshield
Illumulet	Necklace	Produces light when worn
Inquisitor	Sword	Mana plus 2 points
Jewel Symal	Necklace	Anti-magic plus 15 points
Lock Picks	?	?
Mace of Order	Mace	Increases load 4 kgs.
Moonstone	Necklace	Mana plus 3 points
Pendant Feral	Necklace	Wizard levels increased
PowerTowers (Kid's game)	Leg Plates	Strength + 10 points - Load + 8 Kgs.
Rope	Rope	Increases Ninja levels (quickly)
Sceptre of Lyf	Sceptre	Mana plus 5 points
Snake Staff	Staff	Mana plus 8 points
Staff	Staff	Mana plus 2 points
Staff of Manar	Staff	Mana plus 10 points
Storm Ring	Ring	Lightning (limited)
The Conduit (Kid's game)	Staff	Mana plus 16 points
The Hellion	Necklace	?
Vorpal blade	Sword	Mana plus 4 points and Wizard levels
Yew staff	Staff	Mana plus 4 points

Dungeon Skills

Bash	Fighter	
Berzerk	Fighter	
Block	Fighter	Shield
Chop	Fighter	
Cleave	Fighter	
Jab	Fighter	
Melee	Fighter	
Parry	Fighter	
Slash	Fighter	
Stun	Fighter	
Swing	Fighter	
Thrust	Fighter	
War Cry	Fighter	
Climb	Ninja	Builds this level very quickly
Hit	Ninja	Shield
Kick	Ninja	
Punch	Ninja	
Shoot	Ninja	
Stab	Ninja	
Throw	Ninja	
Blow Horn	Priest	Opponent will back away
Brandish	Priest	Works well against flying enemies
Disrupt	Wizard	Vorpal Blade

Softkey for...

Dungeon Master

FTL

Requirements:
Apple IIgs
Copy II Plus v. 9.1

My thanks to Jim Ross. I love Dungeon Master but have been frustrated by the complex and novel arrangement of codes. I would like to add to what he has provided by supplying some additional information. First I have discovered two versions of Dungeon Master. They are 2.0 and 2.1. I also found that the byte locations provided by Jim were not the same for my disk even though I believe that he was working with v 2.0.

For version 2.0

Blk	Byte	From	To
104	110	18	38
104	114	38	18

For version 2.1

Blk	Byte	From	To
0E6	110	18	38
0E6	114	38	18

PS. ⓟ If anyone can find the LOCK PICKS in the game let me know. They are located on the program disk at:
v2.0 Block 558 Track 41 Sector 00 Side 2
v2.1 Block 53A Track 3F Sector 03 Side 2

Charles R. Haight — WA

Editorial Notes

USPS and mail forwarding

Once or twice each month, I receive a call or letter stating "I have not received my issues." Further conversation/reading reveals that a change of address has occured. Somehow they neglected to let Computist know. Hey, I understand! Moving is a busy, confusing and, all to often, traumatic experience. Change of address notices aren't always very high on the list of things to do, although they should be. (And there's always the possibility that you don't want to let people know where you've gone.)

For your information and to clear up why I always ask for postage costs in these instances, let me explain a little about how the U.S. Postal Service (USPS) handles your mail. Providing that you have let them know where you are going, the USPS will do the following, automatically:

1st Class mail is forwarded for one (1) year after which it is returned to sender. 2nd Class mail is forwarded for two (2) months. 3rd Class mail is thrown away. 4th Class mail, provided it looks valuable, is returned to sender.

If you paid $24 for 8 issues then you have a regular subscription. Regular subscriptions are sent by 3rd Class mail. If you moved and didn't inform Computist, then your issues are decomposing in some landfill. That's a waste of money.

In order to have your 3rd Class mail forwarded, you must ask for and sign a form that states that you want your 3rd Class mail forwarded and that you guarantee to pay any forwarding postage. **If you do not ask for and sign this form, your Computist issues are thrown away** along with any other 3rd Class mail. And when you call me and ask for more issues, the least that I will ask for is the 1st Class postage to mail them to you. If there are more than one or two issues, I will charge you the back issue rate.

Since most junk mail is sent by 3rd Class mail, you may not want to pay the postage to have it forwarded to your new

address. The solution is simple, let Computist know when you move and I will send your issues to the correct address. Please, I'm not a mean person and it bothers me to take a hard line. Let me know when you move, it will save us both a lot of headaches.

While we're on the subject of mail, I've got a couple of other comments. If you are changing from a regular sub ($24) to a 1st Class sub ($34), you can send $1.25 for each of the remaining issues of your regular sub and I will upgrade those issues. That way you start getting your issues by 1st Class mail, right away.

Also, if you're late renewing and you know that you missed an issue, send an extra $1.25 for postage and I will back date your renewal and send you the missed issue by 1st Class mail.

Charge It!

I receive a handful of calls each month that ask if we take credit cards. Seems they didn't notice that there is a place on the front cover subscription form for a charge card number and signature. The answer is yes, we take VISA and Master Card. It makes it easier to renew. So if writing a check and addressing an envelope is to much like work, just give us a call and charge it!

IRS?

Here's the latest update on the Computist Chapter 11 action. Due to a clerical error one of the creditors wasn't given sufficient notice, so the final decision on acceptance or rejection of our plan was delayed (again) until January 10, 1991.

The IRS is insisting that we change the plan to allow them to charge us penalties and interest despite the laws that say a chapter 11 estate doesn't pay interest. The Computist lawyer says that the IRS almost always does this because they figure that paying them (the IRS) is cheaper than the legal fees to fight them. Also, there's some question as to whether allowing the IRS to receive interest might also allow other creditors to sue for interest and penalties. Like all legal actions, it's a bucket of worms that us mortals will never really understand. The lawyer say's that paying the IRS in full will simplify things tremendously so I say we go for it.

If you aren't on a limited budget, send a donation to the IRS Fund. Let's pay them off so we can say good riddance and get on with publishing the next issue! Amen.

Non-Subscriber?

If you are calling for information or help, be sure that you tell me that you are not a subscriber. (It won't make any difference in the quality of assistance, I try to help everyone who asks.) When talking on the phone, I don't write down the full address that you give me. I only note the name and state. I figure that I can get the full address from the computer. This causes problems if you are not a subscriber and therefore not listed in the database. I have several letters sitting in my out-box that don't have any addresses on them. If you called and asked for information and you didn't receive a reply, maybe you should call again and make sure I get your address.

Hardware?

Yea, I'm still hoping to get you interested in some kind of hardware project. I made the "Instant On" battery backed SRAM card. It's a lot of fun and really convenient. But, now that I've had some time to play with it, I've thought of some changes that I want. First, there are 12 - 32K SRAM chips on the plug-in board. This comes to 384K of memory. It's not enough! Programs stored on the card are available instantly, so you start putting more of them on the card until you suddenly run out of space. Then you have to pick and choose which ones you need the most. But the worst thing is that you get used to the instant load programs and it seems to take forever when you have to get something from a floppy disk.

The price of 32K SRAM chips is down to $5 each. I've decided that it would be better to design a card(s) that fits into a old floppy disk case. That would allow me to put more 32K chips on the board(s). I figure around 32 chips (1Mbyte of SRAM) should keep me satisfied for a while. That's only $160 for the SRAM chips. What do you think?

Apple II Supercharger

I'm running out of room here too but I wanted to say a few more words about the Option card. (One of our volunteers calles it the IIGTS card which translates as "II Good To Stop.) The circuit design is completed (on paper) and has been tested using an electronic CAD program (DesignWorks from Capilano Computing) with the clock set to 4Mhz. I think that the best way to make the actual board would be to make a plug-in card for slot 7 with three horizontal connectors so that 3 other boards could be plugged into this card. Two of these additional cards would be the CPU card and a memory card. The slot 7 card would have interface circuitry to talk to the Apple.

The reason for the multi-card design is to allow easy (cheap) upgrades to faster CPUs or larger RAM cards. The estimated cost of the project is $250-300. What do you think?

The End of Year Book (loose-leaf, 3-hole punched)

Hey! I'm almost thru with the formats and paging for the 8.5" x 11", laser printed book. I'll be working on it during Christmas vacation. If your tabloid issues are getting as tattered and yellowed as mine then you know why I'm in a hurry to get the book done. Look for an announcement (hopefully) in issue #78 or #79.

IBM RDEX IBM RDEX

Marc Batchelor _____ FL

IBM Softkey for...
Indianapolis 500 "The Simulation"
Electronic Arts

Requirements:
Debug
Indianapolis 500 Original Disk

The protection used on this program is really a "manual" protection. It is however, very good. The program loads a digitized graphic that matches 30 or so in the manual, and asks a question based on the caption underneath. It is a 'cute' implementation of a "manual" protection, but a pain in the tush each time you load the program.

The first thing I tried to do is to search all the files for one of the questions asked. This led nowhere. The text (I assume) is encrypted on the disk. The next thing I did was have a look at the loading program INDY.EXE. The protection is located in the first 64K bank of the program. To locate it, I searched for interrupt 21, function 6 (direct console I/O), parameter FF (input request). I found only one in the first 64K bank. After playing around with this routine, I found that I could make it ignore the results of the input with a few changes. But, I wanted to disable the call all together.

Going a little further, I discovered a CALL to this location. After NOPing the CALL, I executed the program. The program loaded the digitized graphic, displayed it briefly, and then cleared the screen to the actual program. I decided that I wanted to avoid the loading of the graphic as well, so I directed my Debug session into a file. That is, DEBUG file name > FILE.DIS [CR] U100 FFFF [CR] Q [CR]. (Note: You MUST have a large hard drive to accomplish this feat.) I ended up locating a CALL BE9E which called the location that I found (after loading the graphic). I NOP'ed the call, and had a disk that worked fine without ever loading or displaying the digitized graphic and did not ask the player for any information.

Step-by-step

1. COPY INDY.EXE OLDINDY.EXE
2. RENAME INDY.EXE INDY.TMP

```
DEBUG INDY.TMP
S100 FFFF E8 CC 00 E8 B9 8C   Should respond:
   XXXX:YYYY
EYYYY 90 90 90   replace YYYY with previous response
W                              write the changes
```

3. RENAME INDY.TMP INDY.EXE

You are done!

S.D.R.

IBM Softkey for...
Ultima V
Origin Systems

My hat goes off to Origin Systems for not only creating a great adventure in Ultima V, but also splicing in an extra bonus, for people like me, in the form of copy-protection. Although I despise copy-protection I like to unravel its mystery if I can. I also learn from the experience which is good. Undoubtedly Origin could have made a full proof copy-protection but they choose not to. Again thanks to Origin for the double adventure.

1. Copy ULTIMA.EXE from the original program disk to a blank formatted disk. Make sure you have a path to DEBUG or copy DEBUG to the formatted disk. Put the original Ultima program disk away, you won't need it again.
2. From the A:> prompt type:

```
ren ultima.exe ultima.xex
debug
N ultima.xex
L
e 8b65
a5 35 35
e 8baa
a5 35 35
e 8c64
1d 87 a6 06 bd a1 1d 88 a2 06 bf a1
e 8c70
1d a5 ad 06 b9 a1 1d 9e be 06 bb a1
e 8c7c
28 bb bd a6 1c a0 a5 35 35 35 35
w
q
ren ultima.xex ultima.exe
```

Your ULTIMA.EXE will now be unprotected. You can put ULTIMA.EXE on your hard drive or on a copy of the original program disk. When you run the program you will not be asked to place the original disk in drive A:

Mr. Magic _____ PA

IBM Softkey For...
1DIR.UNP
Bourbaki

1DIR is a solid hard disk "manager" program from Bourbaki, Inc. and licensed to Basic Time, etc. to be included with their hard disk units. The copy I received with my unit was "protected" and only ran on the Basic Time hard disk. The following patch will free the software for use on all hard disks.

Copy 1DIR.COM to a diskette or the hard disk. Make sure you have a copy of DEBUG on the same volume. I assume you know how to use DEBUG.

```
DEBUG 1DIR.COM
U3D53
```

Confirm that the line in part reads...

```
"3d53 ... CMP AL, [...]"
```

If yes, this patch will work.

```
A3D53
3D53      XOR AL,AL
3D55      NOP
3D56      NOP
3D57
W
Q
```

That's it. 1DIR should now work on all of your equipment.

By the way, at $99.00 this is a pretty good program—don't rip it off. Unprotect it, if you have other disks to use it with.

NOP (No Obtrusive Protection

IBM Softkey for...
Microstation v3.0D
?

I'm sure some of you, like I do, have problems with those parallel port protection keys that come with some software. When you have 3 or 4 of them attached to the back of your computer, you have to have 8-10 inches (!) of clearance for the back of your machine. And they tend to suffer great harm when attached to a laptop or portable.

I like what Intergraph/Bentley Systems did in making Microstation run as a demo version if the key is not attached. However, while you don't lose your work, the abrupt program exit every 10 minutes or so is kinda annoying.

So I offer a better solution. If this patch works, MicroStation will run continuously without the key. If not, well, sorry to get your hopes up.

I recommend you do this on a copy of your MGDS.EXE file so if something screws up you won't have to re-install Microstation.

```
copy mgds.exe *.old   make a backup just in case
ren mgds.exe x        rename file for DEBUG
debug x               DEBUG loads file
E 5BC9 EB             patch file
E 9BA2 EB
E B127 EB
R                     Display registers
```

Look at the entry CS=xxxx, where xxxx is some hexadecimal number. Add 3000 hex to it to get another hex number yyyy. (ie. If xxxx=12A9, then yyyy=42A9.)

```
E yyyy:3C68 B8 04 45 89 46 FC EB   use correct
   yyyy value
W                     mmmmm bytes written write file
Q                     quit DEBUG
ren x mgds.exe        restore original name
```

Done! Now run Microstation, and hope all goes well. Good luck.

UNK

IBM Softkey for...
Where in Time is Carmen Sandiego
Broderbund

This program checks for the original key disk each time you are promoted in the game.

I found another softkey (CARMNTME .ZIP) which did not work at all. If fact, it made it so you can not even get into the program!

Here's mine. You can patch the file either with a hex editor, or with debug.

Using a hex editor

Search for 02 E1 07 C3 FA 55 and change the 02 to 12.

Using debug

```
ren carmen.exe car
debug car
S0000 FFFF 02 E1 07 C3 FA 55
xxxx:yyyy             note value of yyyy
e yyyy                type e, then value of yyyy above
xxxx:yyyy 02.12
w
q
ren car carmen.exe
```

That's it. And a lot easier then the other patch that didn't work! Although this patch only changes one byte, it was difficult to come up with. They used a far call which was hard to follow, and could not be nop'ed out. Good Luck.

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

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

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

Wanted Apple IIe Software

Especially Wargames and science fiction. Also, what hardware do you have? Sell:

Epson APL board (no docs) $12
SpeedDemon Accellerator card (3.5Mhz) $60
Apple IIe 80 column card (only) $10
Apple IIe 80 column card (*/64K) $20
Dumpling-GX printer card (no docs) $12
Apparate PROM blasting System $25
Apple II+ motherboard (works OK) $60
Grappler clone printer card $20

I also have a few Apple One items, send SASE for more info.

Joe Torzewski
51625 Chestnut Road
Granger IN 46530

$10 Ten Buck Sale $10

All Software Only $10 Each
Black Cauldron — IIe
Copy II Plus v8.3 (disk only) — IIe
Curse of the Azure Bonds — IIe
Destroyer — IIe
F-15 Strike Eagle — IIe
Into the Eagle's Nest — IIe
Karate Champ — IIe
Marble Madness — IIe
Might & Magic — IIe
Print Shop — IIe
Russia: The Great War in the East — IIe
Softdisk Sampler — IIgs

Send Check/Money Order to:

Alan T. Zak
9724 N.W. 59th Place
Gainesville, FL 32606

RENT or BUY IIgs SOFTWARE

- Money Back Guarantee
- Catalog contains over 200 titles
- Rent for 15% - 20% of list price
- Prices comparable with mail order
- Rental fee discounted from purchase price
- $15 Lifetime Membership Fee

GSoft
7350 Ulmerton Road - Suite 924
Largo, Florida 34641
(813) 536-4352

RDEX Contributors:

The CPR	Agent	5
Marc	Batchelor	8, 22
Roland	Boucher	11
Alan	Chaney	9
Dr	Crack	12
Rick	Davis	14
	Edison	4
Edward	Eastman	17
Gary	Wills	18
Kim	Griffith	18
Suzanne R.	Harvey	20
E.N.	Hondrick	11
Steve	Kalynuik	10
J. L.	Walters	14
Ed	Lee	4
Robin	Locksley	21
Mr.	Magic	22
	Momma	13
Rod	O'Brien	13
NOP (No Obtrusive	Protection	22
S.D	R.	22
George	Sabeh	16
Ephraim	Santiago	5
	UNK	22

Most Wanted

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
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
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	Flt 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
69	Hard Hat Mack	?
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 Developement
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 Level 5 and 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	Phantasie 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
67	Rear Guard	Adventure International
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
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
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 (IIe)	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
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
61	Wasteland	Electronic Arts
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

73 The Product Monitor - Dragon Wars - Fast Frames, Updates, etc. - 2088: Bargain Thrills - Dragon Wars Tavern Tales •Features, Notes and such: A note about Reading & Me •Autocopy Parm to copy Ultima V: •BBS News (RDEX entries) •BBS Notes (Questions and Answers) •Behind the scenes of a One Byte Patch •Blocks vs Tracks — ProDOS & DOS Converting from/to blocks & sectors •Bug in Crystal Quest Mod (issue #71) •Cheap NMI? •Comments on Assembly Language Programming for Beginners •Comments: Copy II Plus 9.0 •Free Adventure Maps •MONRWTS A Read/Write disk routine •Note on Bilestoad •Notes on Disk Muncher •Notes on LaserForce •Notes on War in Middle Earth •Put Ancient Land of Ys on a Harddrive •Putting Shanghai on a hard disk •Quick and dirty ProDOS 16 READ_BLOCK patch •Rick's Hello •Run Paintworks Gold under GS/OS 5.0 •Senior PROM — NOT Available •Some notes on Copy II Plus v9 •Some notes on RISK (Leisure Games) •Ultima II Character Editing •Ultima IV HELP •Using Copy II plus to make an APT •Softkeys: Ancient Land of Ys •Arkanoid II: Revenge of Doh •Bad Dudes •Battle Chess •Battle Chess GS •Bubble Ghost GS •Charlie Brown's ABC's •Chem Lab •Curse of the Azure Bonds (IIe) •Downhill Challenge •Four-in-One Infocom Sampler •GBA Championship Basketball •Geometry v1.0 •Grand Prix Circuit •Great Western Shootout •Heavy Barrel •How the West was One + Three x Four •Impossible Mission II •Jack Nickalaus' Greatest 18 Holes of Major Championship Golf •King of Chicago •Neuromancer •Operation Frog •Planetfall •Platoon •Pool of Radiance (IIe) •Sokoban •Stickybear Opposites GS •Stickybear Shapes GS •Study Skills •Test Drive II: The Duel •Test Drive II: The Duel (GS) •The Children's Writing and Publishing Center •The Design Your Own Home Series -Architectural Design -Interior Design -Landscape Design •Think Quick v1.2 •Three Stooges •Tunnels of Armageddon •U.S.A. Geograph v1.0 •Ultima II (Original Version) •Where in North Dakota is Carmen Sandiego •World Geograph v1.1 •Bitkeys: •Dungeon Master •Gradebusters 123 v3.35 •Math Blaster Plus! v3.1 •Strike Fleet •Where in the USA is Carmen Sandiego •APTs: A.E. •Alien Ambush •Alien Game •Apple Kong •Apple Panic •Battle Zone •Beer Run •Bellhop •Berserker •Bolo •Borg •Bruce Lee •Buck Rogers •Bug Attack •Buzzard Bait •Cannonball Blitz •Canyon Climber •Captain Power •Caverns of Calisto •Ceiling Zero •Choplifter •Color Planetoids •Congo •Creepy Corridoors •Crisis Mountain •Crossfire •Crystal Castles •Cyclod •Diamond Mine •Dig 'Em •Dig Dug •Donkey Kong •Dragon Wars •Drelbs •Drol •Dung Beetles •Electro Arena •Eliminator •Evolution •Falcons •Falcons II •Fire and Ice •Free Fall •Frogger •Galaxian •Genetic Drift •Gobbler •Gold Rush •Halloween •Hard Hat Mack •Hellstorm •Hellstrom •Horizon V •Ice Demons •Jawbreaker II •Joust (Atarisoft) •Jouster •Jump Jet •Jumpman •Kameari •Labyrinth •Mapple: •Marauder •Mars Cars •Microwave •Milipeed •Miner 2049er •Money Munchers •Montezuma's Revenge •Mouskattack •Mr Cool •Mr. Robot and the Robot Factory •Neptune •Night Crawler •Nightmare Gallery •Nomads •Oils Well •Outpost •Phaser Fire •Pooyan •Quadrant 6112 •Quest For Tires •Raiders of the Lost Ring •Randamn •Raster Blaster •Rearguard •Repton •Ribbit •Robotron: 2084 •Sammy Lightfoot •Sea Dragon •Sea Fox •Serpentine •Situation Critical: •Snack Attack •Snake Byte •Snapper •Sneakers •Snoogle •Space Cadets •Space Quarks •Spy's Demise •Star Maze •Star Thief •Star Trek •Succesion •Super Puckman •Swashbuckler •Syzygy •Taxman •Teleport •Thief •Threshold •Thunderbombs •Tubeway •Tubeway II •Viper •Wargle •Warlock GS •Zany Golf GS •Playing Tips: Bard's Tale I •Dungeon Master •Neuromancer •Ultima III •Ultima V •Where in the World is Carmen SanDiego •Wizardry (Proving Grounds of the Mad Overlord) •Wizardry •Zork Zero •**IBM** Playing Tips: Mean Streets •Mech Warrior •Pool of Radiance

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 licencee'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 •Designasauraus •Dig Dug •Duel (gs) •Early Games •Eliminator •Equation Math •Estimation •Explora-Classic series •Explora-Science Whales •Fraction Concepts •Fraction Practice Unlimited •Fraction Recognition •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 •Guiness World Records •Homeworker •Houghton Mifflin Math Courseware •Ice Demons •Ivitation 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 SanDiego •Where in the World is Carmen SanDeigo? (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 65¢ stamp to Computist.

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

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 8½ 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

	US, Canada & Mexico	All others
Back issues (1 - 9)	$4.75	$8.75
Back issues (10+)	$3.75	$6.00
Zox back issues	$4.75	$8.75
Library disks (1 - 9)	$5.50	$7.50
Library disks (10+)	$4.00	$6.00

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 5¼ 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 65¢ 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 _____ CP77

• 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