For The Serious User Of Apple][Computers

COMPUTIST

Issue No. 30 $3.75

Softkeys For:

Millionaire
SSI's RDOS
Fantavision
Spy vs. Spy
Dragonworld
King's Quest
The Bard's Tale
Space Shuttle

Core:

Ultimaker IV: An Ultima IV Character Editor

Feature:

Increasing Your Disk Capacity

(Page 11)

COMPUTIST
PO Box 110846-T
Tacoma, WA 98411

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

Many of the articles published in COMPUTIST detail the removal of copy protection schemes from commercial disks or contain information on copy protection and backup methods in general. We also print bit copy parameters, tips for adventure games, advanced playing techniques (APT's) for arcade game fanatics and any other information which may be of use to the serious Apple user.

COMPUTIST also contains a special CORE section which focuses on information not directly related to copy protection. Topics may include, but are not limited to: tutorials, hardware/software product reviews and application and utility programs.

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

Commands And Controls: In any article appearing in COMPUTIST, commands which a reader is required to perform are set apart from normal text by being indented and bold. An example is:

PR#6

Follow this with the RETURN key. The RETURN key must be pressed at the end of every such command unless otherwise specified.

Control characters are indicated by being boxed. An example is:

6 CTRL-P

To complete this command, you must first type the number 6 and then place one finger on the CTRL key and one finger on the P key.

Requirements: Most of the programs and softkeys which appear in COMPUTIST require one of the Apple][series of computers and at least one disk drive with DOS 3.3. Occasionally, some programs and procedures have special requirements. The prerequisites for deprotection techniques or programs will always be listed at the beginning of the article under the "Requirements:" heading.

Software Recommendations: The following programs (or similar ones) are strongly recommended for readers who wish to obtain the most benefit from our articles:

1) **Applesoft Program Editor** such as Global Program Line Editor (GPLE).
2) **Sector Editor** such as DiskEdit, ZAP from Bag of Tricks or Tricky Dick from The CIA.
3) **Disk Search Utility** such as The Inspector, The Tracer from The CIA or The CORE Disk Searcher.
4) **Assembler** such as the S-C Assembler or Merlin/Big Mac.
5) **Bit Copy Program** such as Copy][Plus, Locksmith or The Essential Data Duplicator
6) **Text Editor** capable of producing normal sequential text files such as Applewriter][, Magic Window][or Screenwriter][.

You will also find COPYA, FID and MUFFIN from the DOS 3.3 System Master Disk useful.

Super IOB: This program has most recently appeared in COMPUTIST No. 22. Several softkey procedures will make use of a Super IOB controller, a small program that must be keyed into the middle of Super IOB. The controller changes Super IOB so that it can copy different disks. To get the latest version of this program, you may order COMPUTIST No. 22 as a back issue or order Program Library Disk No. 22.

RESET Into The Monitor: Some softkey procedures require that the user be able to enter the Apple's system monitor during the execution of a copy protected program. Check the following list to see what hardware you will need to obtain this ability.

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

Apple][Plus - Apple compatibles: 1) Install an F8 ROM with a modified RESET vector on the computer's motherboard as detailed in the "Modified ROM's" article of COMPUTIST No. 6 or the "Dual ROM's" article in COMPUTIST No. 19.

Apple //e - Apple //c: Install a modified CD ROM on the computer's motherboard. Clay Harrell's company (Cutting Edge Ent.; Box 43234 Ren Cen Station-HC; Detroit, MI 48243) sells a hardware device that will give you this ability. Making this modification to an Apple //c will void its warranty but the increased ability to remove copy protection may justify it.

Recommended Literature: The Apple][Reference Manual and DOS 3.3 manual are musts for any serious Apple user. Other helpful books include: *Beneath Apple DOS*, Don Worth and Peter Leichner, Quality Software, $19.95; *Assembly Language For The Applesoft Programmer*, Roy Meyers and C.W. Finley, Addison Wesley, $16.95; and *What's Where In The Apple*, William Lubert, Micro Ink., $24.95.

Keying In Applesoft Programs: BASIC programs are printed in COMPUTIST in a format that is designed to minimize errors for readers who key in these programs. To understand this format, you must first understand the formatted LIST feature of Applesoft.

An illustration- If you strike these keys:

10 HOME:REMCLEAR SCREEN

a program will be stored in the computer's memory. Strangely, this program will *not* have a LIST that is exactly as you typed it. Instead, the LIST will look like this:

10 HOME : REM CLEAR SCREEN

Programs don't usually LIST the same as they were keyed in because Applesoft inserts spaces into a program listing before and after every command word or mathematical operator. These spaces usually don't pose a problem except in line numbers which contain REM or DATA command words. The space inserted after these command words can be misleading. For example, if you want a program to have a list like this:

10 DATA 67,45,54,52

you would have to omit the space directly after the DATA command word. If you were to key in the space directly after the DATA command word, the LIST of the program would look like this:

10 DATA 67,45,54,52

This LIST is different from the LIST you wanted. The number of spaces you key after DATA and REM command words is very important.

All of this brings us to the COMPUTIST LISTing format. In a BASIC LISTing, there are two types of spaces; spaces that don't matter whether they are keyed or not and spaces that must be keyed. Spaces that must be keyed in are printed as delta characters (Δ). All other spaces in a COMPUTIST BASIC listing are put there for easier reading and it doesn't matter whether you type them or not.

There is one exception: If you want your checksums (See "Computing Checksums" section) to match up, you *must not* key in any spaces after a DATA command word unless they are marked by delta characters.

Keying In Hexdumps: Machine language programs are printed in COMPUTIST as both source code and hexdumps. Only one of these formats need be keyed in to get a machine language program. Hexdumps are the shortest and easiest format to type in.

To key in hexdumps, you must first enter the monitor:

CALL -151

Now key in the hexdump exactly as it appears in the magazine ignoring the four-digit checksum at the end of each line (a "$" and four digits). If you hear a beep, you will know that you have typed something incorrectly and must retype that line.

When finished, return to BASIC with a:

E003G

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

Keying In Source Code The source code portion of a machine language program is provided only to better explain the program's operation. If you wish to key it in, you will need an assembler. The S-C Assembler is used to generate all source code printed in COMPUTIST. Without this assembler, you will have to translate pieces of the source code into something *your* assembler will understand. A table of S-C Assembler directives just for this purpose was printed in COMPUTIST No. 17. To translate source code, you will need to understand the directives of your assembler and convert the directives used in the source code listing to similar directives used by your assembler.

Computing Checksums Checksums are four digit hexadecimal numbers which verify whether or not you keyed a program exactly as it was printed in COMPUTIST. There are two types of checksums: one created by the CHECKBIN program (for machine language programs) and the other created by the CHECKSOFT program (for BASIC programs). Both programs appeared in COMPUTIST No. 1 and The Best of Hardcore Computing. An update to CHECKSOFT appeared in COMPUTIST No. 18. If the checksums these programs create on your computer match the checksums accompanying the program in the magazine, then you keyed in the program correctly. If not, the program is incorrect at the line where the first checksum differs.

1) To compute CHECKSOFT checksums:

LOAD filename
BRUNCHECKSOFT

Get the checksums with

&

And correct the program where the checksums differ.

2) To compute CHECKBIN checksums:

CALL -151
BLOAD filename

Install CHECKBIN at an out of the way place

BRUN CHECKBIN,A$6000

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

xxx.xxx CTRL-Y

And correct the lines at which the checksums differ.

Coping with COMPUTIST

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

Our editorial policy is that we do NOT condone software piracy, but we do believe that honest 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 application programs to meet his or her needs.

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

renew your freedom

Check your mailing label to see if you need to renew your subscription. And if you think you might forget when that fatal time arrives, renew right now. Just use the order blank below.

if you're moving...

Let us know right away or at least 30 days in advance so that you won't miss a single issue. Just write your new address below, and include your present address label. Issues missed due to non-receipt of this Change-of-Address may be aquired at the regular back-issue rates.
Please remember, the Post Office does not forward third class mail unless requested.

We are NOT PIRATES

but we're not fools, either.

We're serious programmers and software users who just want to have backup copies of any software we own. COMPUTIST magazine shows us HOW TO MAKE BACKUPS OF COMMERCIAL SOFTWARE regardless of the maker's attempt to stop us from having legal copies. Don't let them stop you from protecting your own rights.
Remove copy-protection from your valuable library of expensive software. The publisher of COMPUTIST has been showing subscribers how to unlock and modify commercial software for the past 4 years. Don't be one of the users abused by user-FIENDLY locked-up software. Subscribe.

6 issue SUBSCRIPTION RATES:
U.S.: $20 first class: $24
Canada, Mexico: $34
Other Foreign: $60

SAMPLE COPY:
U.S.: $4.75 Foreign: $8.00

US funds drawn on U.S. bank.
In Washington add 7.8% tax.
Send check or money order to:

COMPUTIST
PO Box 110846-T
Tacoma, WA 98411

☐ NEW subscriber ☐ Renew my subscription ☐ New address

Name _____ ID# _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

Check Money Order _____-_____-_____-_____ Exp. _____

Signature _____ CP30

ATTENTION ADVENTURERS

Adventure Tips .....and Clues

COMPUTIST is looking for more adventure hints to any of the popular adventure/fantasy games sold for the Apple II, II Plus, or //e. These will be used in our regular column, ADVENTURE TIPS.

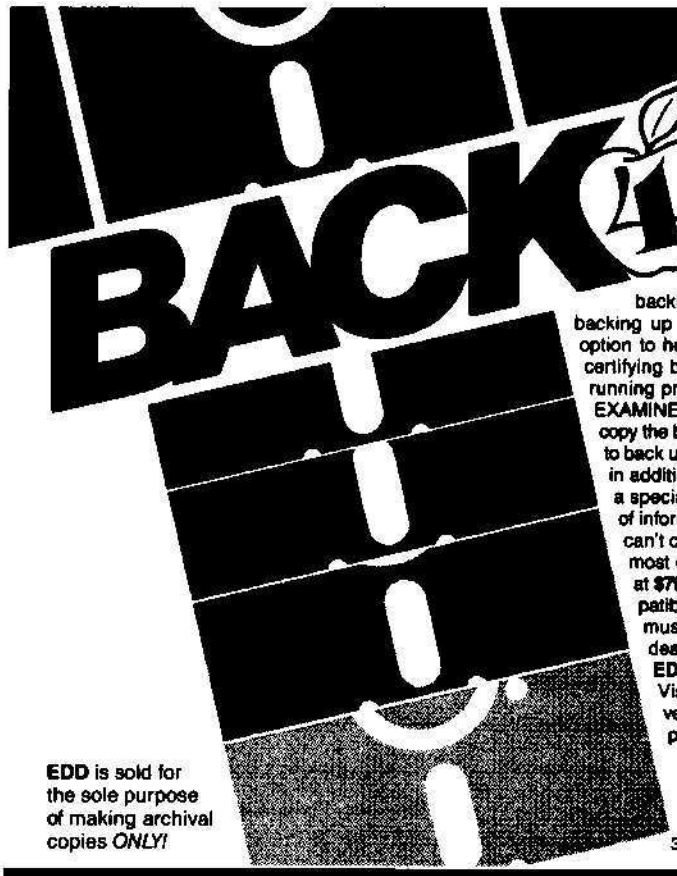
We prefer that these hints not be a dead giveaway to solutions of dilemmas presented by the particular game. Adventure tips should contain just enough information to nudge the stumped adventurer towards the answer.

How & Where

So, if you know how to open the jewel-encrusted egg, how to plug the hole on the rowboat, where to find the key to the treasure chest, or any other tidbits of information that may be helpful to your fellow traveler, please send this information on a 3 × 5 postcard to:

COMPUTIST TIPS
PO Box 110846
Tacoma, WA 98411

P.S. Please don't forget to include the name of the adventure game to which your hint pertains and the name of the manufacturer. The grateful sighs of all the readers you have helped shall be your just payment. Maybe your Charisma will go up, too!



BACK UP YOUR DISKS

EDD Version 4 is the most powerful copy program available for backing up "uncopyable" or "copy-protected" disks. ■ In addition to backing up disks, **EDD 4** also features a hi-resolution graphic DISK SCAN option to help you locate information on a disk, a CERTIFY DISK option for certifying blank disks, and since it's very important that your disk drives are running properly (especially when copying disks), we have also included an EXAMINE DISK DRIVE option. ■ Even though **EDD 4** has been preset to copy the broadest range of copy-protections possible, **EDD 4** can be "modified" to back up almost any disk that runs on your Apple! ■ For the dedicated user, in addition to **EDD 4**, we are offering an **EDD 4 PLUS** version that includes a specially designed hardware card which allows **EDD** to copy EVERY bit of information from each track accurately! You can bet that if **EDD 4 PLUS** can't copy it, nothing will! ■ **EDD 4** runs on an Apple II, II Plus (including most compatibles, IIe, IIc, and III (using emulation mode), and is priced at **$79.95**. ■ **EDD 4 PLUS** runs on Apple II, II Plus (including most compatibles), and IIe, and is priced at **$129.95** (duodisk/unidisk 5.25 owners must add $15 for a special cable adapter). Ask for **EDD** at your local dealer, or to order direct, include $3 ($6 foreign) shipping/handling for **EDD 4**, or include $5 ($8 foreign) for **EDD 4 PLUS**. ■ Mastercard and Visa accepted. All orders must be prepaid. ■ If you have an earlier version of **EDD**, you can update to **EDD 4** or **EDD 4 PLUS** at a reduced price. Send your **EDD** disk to us, and deduct $50 from your order.

EDD is sold for the sole purpose of making archival copies *ONLY!*

UTILICO MICROWARE

3377 SOLANO AVENUE / SUITE 352 / NAPA, CA 94558 / 707-257-2420

ESSENTIAL DATA DUPLICATOR 4

Unlock the Software Secrets

The Senior PROM Version 2.0

Allows your Apple //c or //e to **halt** a program and:

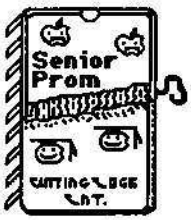
- Instantly swap between two different 64k programs.
- Examine where in memory the program is running.
- Disassemble any memory, even volatile $200-7FF.
- Examine the Stack for return subroutine addresses.
- Alter any memory to examine running effects.

The program may then be **restarted** exactly where it was interrupted or at any other location, or be **saved** to disk in normal B-files and later restarted.

Also includes a sophisticated Sector Editor and Memory/Disk Detective, and operating system with disk copy, format, edit, and protected disk utilities, all **without** booting DOS first! Assembly Language utilities include Step & Trace, an Assembler, & more.

All the **Senior PROM's** utilities are in **ROM**, instantly available when needed. Undetectable by any software or hardware, does not use a peripheral slot! Extensive documentation & guide to copy protection.

Economically priced at **$79.95** for prepaid orders with check or money order. Credit card orders available for **$88.95**. Please specify //c, or //e Standard or Enhanced ROMs.



Cutting Edge Enterprises
Box 43234 Ren Cen Station
Detroit, MI 48243

For VISA orders call
517-743-4041, 10-5 E.S.T.
or 313-349-2954 Modem 24 hrs.

The Doctor Is In!

Presenting *Snoopy and the Doctor (SanD)*

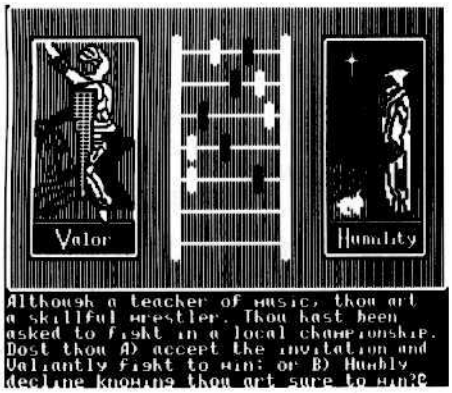
An easy to use utility that provides disk and memory edit, search, and viewing functions. Also includes disk compare, nibble read, half track support, decimal/hex converter, free space map, file follower and more.

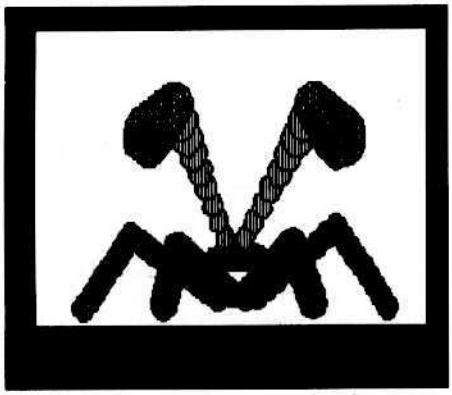
SanD is unlocked and easily modifiable. "Merlin"™ source code included allowing user modifications and re-assembly at ANY location in memory. Includes lower RAM and RAM card versions.

For all Apple // computers. Disk version **$24.95**. Also available in two 2716 **EPROMs**: add **$10.00**. Check or Money Orders only please.

Cutting Edge Enterprises
Box 43234 Ren Cen Station
Detroit, MI 48243
313-349-2954 Modem 24 Hours

EPROM Burning Service: We can burn your 2716, 2732, 2764 or 27128 EPROMs. Send blank EPROM, code to burn (on Apple // disk), and $12 per EPROM. Over 4, $10 per EPROM.





This month's cover:
Graphics from Electronic Art's "The Bard's Tale."

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

Return postage must accompany all manuscripts, drawings, photos, disks, or tapes if they are to be returned. Unsolicited manuscripts will be returned only if adequate return postage is included.

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

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

Apple usually refers to an Apple][Computer, and is a trademark of Apple Computers, Inc.

SUBSCRIPTIONS: Rates (for 6 issues): U.S. $20, U.S. 1st Class $24, Canada & Mexico $34, Foreign $60. Direct inquiries to: **COMPUTIST, Subscription Department, PO Box 110846-T, Tacoma, WA 98411.** Please include address label with correspondence.

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

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

COMPUTIST

Issue 30

Publisher/Editor: Charles R. Haight **Managing Editor:** Ray Darrah
Technical Editor: Robert Knowles **Circulation:** Michelle Frank, Debbie Koval
Advertising: (206) 474-5750 **Printing:** Valco Graphics Inc., Seattle, WA
COMPUTIST is published monthly, except December, by SoftKey Publishing, 5233 S. Washington, Tacoma, WA 98409
Phone: (206) 474-5750

softkeys

12 Millionaire
by William Clarke

21 SSI's RDOS
by Mike McConnell

24 Fantavision
by Mike Saul

26 Spy vs. Spy
by Danny Pollak

28 Dragonworld
by Timothy James Strelchun

feature:

27 Increasing Your Disk Capacity
Tired of getting a "DISK FULL" error? This article details two nifty little tricks that will enable you to get the most out of your diskettes. *by Phil Goetz*

core:

16 Ultimaker IV, an Ultima IV Character Editor
In keeping up with the recent release of Ultima IV: Quest of the Avatar, COMPUTIST presents a program that will allow you to create super characters before entering the terrifying dungeons ahead. *by Danny Pollak*

departments

4 Input

6 Most Wanted List

28 Adventure Tips

7 Readers' Softkey & Copy Exchange

Sierra On-Line's **King's Quest** *by Jean Michel George*, CBS Software's **Mastering the SAT** *by Danny Pollak*, Springboard's **Easy as ABC** *by B. Croome*, Activision's **Space Shuttle** *by Jordi Le Vant*, Sunburst Education's **The Factory** *by Doug Walker*, Paladin's **Visidex 1.1E** *by Thomas J. Scott*, Bantam's **Sherlock Holmes** *by Ed Croft*, Electronic Arts's **The Bard's Tale** *by Charles Taylor*

input

Please address letters to:

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

Include your name, address and phone number.

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

Our technical staff is available for phone calls between 1:30 pm and 4:30 pm (PST) on Tuesdays and Thursdays only.

Pronto 65C02

As the author of ProntoDOS, I was much surprised to read in your COMPUTIST No. 27 (page 6) that my sanely great software won't function correctly with the 65C02 chip. Since this would be a rather critical flaw, I immediately asked Beagle Bros' Uncle Louie to do some exhaustive testing. Before sneaking off for a nap, he reported that ProntoDOS worked as well with the 65C02 as it did with the 6502, as far as he could tell.

For the record, my primary goal while writing ProntoDOS was to change as little of DOS 3.3 as possible. The Beagle Bros insisted that Pronto DOS work with all their previous programs, tips, and tricks. They had mucked around in DOS a lot. Consequently, ProntoDOS actually changes fewer that 5 percent of the bytes inside DOS 3.3.

It doesn't remove INIT. It doesn't remove error messages. It doesn't mess with any of the "free" spaces inside DOS. It fully supports all DOS 3.3 commands and RWTS calls. What couldn't be squeezed in, was support for all file manager calls. Consequently, ProntoDOS doesn't work with most programs that call the file manager directly, notably ASCII Express, the Merlin Assembler, and Manx C. On the other hand, unlike most other DOS-speed-up utilities, it does work with almost all imaginable DOS peeks, pokes, and other paraphenalia.

Your subscribers might be interested to know that while sales of the Beagle Bros version of ProntoDOS have fallen off since Apple released ProDOS, the number of software companies with active licenses to use ProntoDOS on their disks have continued to grow. There is still a lot of software being developed and sold under DOS 3.3 rather than ProDOS.

And while I have your attention, how come you've never informed your readers of my sanely great newsletter, Open-Apple. I give your publication free plugs all the time.

Tom Weishaar
Overland Park, KS

Mr. Weishaar: *We had recieved several reports of Pronto DOS not working with the 65C02 and thought it best to relay this information to our readers. We apologize for any inconvenience this may have caused you, as your tests indicate our claims to be false.*

Pronto DOS is truly a great piece of software. Equally as great is your publication "Open Apple." We read it cover-to-cover every month. "Open Apple" is a must for every serious user of Apple][Computers. We especially enjoy your unprotected format.

Open Apple
POB 7651-HC
Overland Park, KS 66207

F-15 Strike Eagle Revisited

I read Mr. Williams's letter in COMPUTIST No. 27 stating that he was having trouble making a backup copy of F-15 Strike Eagle. I also used the controller in COMPUTIST No. 24 and was unable to backup F-15, even using Copy II plus 5.5 which has 3 parms for F-15. I went snooping with my sector editor on different locations that Mr. Jasonowicz stated in his softkey. So I modified his controller and the procedure worked with my copy.

By the way, this game is one of my favorites. My copy of F-15 was purchased as soon as F-15 was released and my Flight Operations Manual indicates technical order # 1-F-15E-1; 15 JANUARY 1985, CHANGE 2.

controller

```
1000 REM FAST CONTROLLER FOR F-15 HC#24
1005 REM MOD ON LOCATION OF EDITS
1010 TK=0 :LT=6 :ST=15 :LS=15 :CD=WR :FAST
     =1
1020 GOSUB 490 :GOSUB 610 :LT=35 :RESTORE :T1
     =TK :TK=PEEK (TRK) - 1 :GOSUB 310 :TK=1
1030 GOSUB 490 :GOSUB 610 :IF PEEK (TRK) =LT
     THEN 1050
1040 TK=PEEK (TRK) :ST=PEEK (SCT) :GOTO 1020
1050 HOME :PRINT "COPYDONE" :END
5000 DATA 6 CHANGES
5010 DATA 31 ,0 ,222 ,234 ,31 ,0 ,223 ,234 ,31 ,0
     ,224 ,234
5020 DATA 33 ,10 ,113 ,234 ,33 ,10 ,114 ,234 ,33
     ,10 ,115 ,234
```

Do any of your readers know where I can purchase a copy of Atari Soft's Pole Position for the Apple][? As far as I can tell, this program is no longer being marketed.

SOLO 7

Strike Eagle Revisited (again)

After reading Mr. Williams's letter in COMPUTIST No. 27 about an updated Strike Eagle, I noticed he had the same type of protection as Silent Service also by MicroProse. Track $22 looks like $21, et cetera. After comparing Larry Jasonwicz's article (COMPUTIST No. 24) on the F-15 Strike Eagle to Silent Service, it seemed MicroProse changed their protection code slightly and moved it. I also found that track 4 sector 9 was the same as track 5 sector 9, and they both contain the protection code.

The disk can be copied with a fast copier such as Locksmith fast copy, don't worry about the error on track $22. All you need to do now is a few sector edits and it's as good as new. One must keep in mind that I can only say this works for Silent Service dated 15-Sept-85, and there is a good chance it will work on the updated F-15 Strike Eagle.

TRACK	SECTOR	BYTE	FROM	TO
4	9	B8	02	DB
5	9	B8	02	DB

(Note: Edits must be done on the copy)

If this doesn't work you can search the disk for a set of codes that looks somewhat like this:

```
C5 2E D0 16 A9 DB 8D 01 02
```

Disassembled:

```
CMP $2E
BNE $02B3 (BNE<= May also be BEQ or BCS)
LDA #$DB
STA $0201
```

Then all you have to do is find out where they load in the BAD value and replace it with the CORRECT value.

Neil Lemme
Highwood, IL

A Couple of Softkeys

Mask Parade

Springboard Software

Requirements:
Apple][series or clone
One or two disk drives
One blank disk
Sector editor program
Mask Parade disk

Mask parade is an excellent program for children of all ages. It not only gives them some computer experience but also gives them something for their trouble. It can also be normalized, a must for this type of program, by using COPYA and a sector editor.

The easy method:

1) RUN COPYA.

2) Break-out with CTRL-C.

3) Enter monitor

Call -151

4) Change B942 from 38 to 18.

5) Re-enter basic

3D0G

6) Run and copy both sides of the disk.

7) Sector Edit Track 00, Sector 03, Byte 42 from 38 to 18.

8) All done and have fun.

Videx 80 col. Pre-boot for Visi-calc and Apple Writer][

Videx Inc.

Requirements:
Apple][Plus or clone
Demuffin Plus on tape
One blank Init disk
Videx pre-boots

Demuffin plus must be loaded from tape as the pre-boot disks wipe out memory when they are booted.

Although I did this one quite some time ago and no longer have the originals to try it, Super IOB with swap controller should work just as well.

In cookbook form:

1) Get Demuffin plus onto tape.

2) Boot one of the pre-boots.

3) Break into the monitor, Apple Writer is the easiest of the two.

4) Check DOS warm start pointer at $3D0 and be sure it points to $9DBF or Demuffin plus will not work.

5) Load Demuffin plus from tape. I used $6000.$8000.

6) Move to $803 to RUN.

803<6000.8000M

7) RUN with 803G.

8a) For Visi-calc, copy one binary file. Mine was Pre Boot 03/22. Do not need to copy Applesoft Pre-Boot program. To use, just BRUN Pre Boot 03/22.

8b) For Apple Writer][, copy 16 files starting with Hello. To use, RUN HELLO program.

Edward Hauff
Red Deer, Alberta, CANADA

Your Personal Net Backup

To make *Your Personal Net Worth* by The Scarborough System into a normal DOS 3.3 unprotected disk, follow the steps below:

1) Use the MasterType Softkey procedures from COMPUTIST No. 15 or the Alternate MasterType softkey from COMPUTIST No. 18 to make a COPYAble version of *Your Personal Net Worth*.

2) Initialize a blank disk using normal DOS 3.3 and with "NW" as the boot program, then copy all files from your softkeyed version to the new disk. Make sure you write-protect the disk before booting.

The DOS used by *Your Personal Net Worth* has altered the commands SAVE, INIT, and CATALOG; to INIT, SAVE, and LAB-BOB; respectively, but these commands do not seem to be used by *Your Personal Net Worth*, so changing them with a disk editor is unnecessary.

If you want a faster booting *Your Personal Net Worth*, load the file "NW" and delete lines 110 through 141. Save the file with the same name.

Joseph Kline
APO New York, NY

Potpourri

After trying unsuccessfully to backup the language disk from my copy of Terrapin Logo v1.2 with several nibble copiers, I tried Super IOB with the standard swap controller. It worked!

As an extra bonus, I found that there was room on the now COPYAble language disk for all of the files from the Utility Disk.

I've also found that the softkey for Spinaker's Story Maker on page 4 of COMPUTIST No. 22 will also work on Delta Drawing, another Spinaker program.

Keep up the good work.

John Cotter
Bay CIty, MI

A Revelation

After deprotecting many disks using the articles in COMPUTIST, I would like to share my first successful solo deprotection. By comparison to some softkeys featured, it's meager, but as a novice with little understanding of machine language, I'm excited! It is for DLM's Alphabet Circus. By using CIA's Tricky Dick/Linguist, I found that the protection used was that of altered marks. They are as follows:

```
D7 AA 96    DE AA
D7 AA AD    DF AA
D7 AA 96    DE AA EB
```

input

After reading through many back issues to compare controllers, I came up with the following. It worked perfectly and my disk is now COPYAble. I used Pronto Dos to INIT the copy and Super IOB version 1.2 to deprotect it. I had tried to use the softkeys for DLM printed previously, but apparently they've changed again.

```
1000 REM ALPHA CIRC
1010 TK = 3 : LT = 35 : CD = WR : MB = 151 : ONERR GOTO
     550
1020 ST = 0 : T1 = TK : GOSUB 490 : RESTORE : GOSUB
     190 : GOSUB210 : GOSUB170
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 : TK = T1 : ST = 0 : GOSUB 490
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 : PRINT "COPYDONE" : END
5000 DATA 215 ,170 ,150 ,215 ,170 ,173 ,222 ,170
     ,223 ,170
```

An now for another matter. I used the Skyfox softkey, and it buzzed right through. It boots and runs, except that when I try to play anything that involves tanks, the sound scrambles. I have a Mockingboard in slot 4, which enhances the sound incredibly. The original works fine. Also, the Space Invaders feature does not function on the copy. Can anyone help???

Lastly, would it be possible to run a complete tutorial, including EXAMPLES, on Super IOB in future issues? I have read the older tutorials and feel they assume a little too much concerning my knowledge of what's going on and particularly WHY!!!!! Case in point: My version 1.2 has worked well on many disks, but the new 1.5 just doesn't fly! A tutorial might just help us limited knowledge people understand a bit more, so that ultimately we could contribute to future issues!

Michael Ferreira
Santa Rosa, CA

Karateka Fix

I see that the "mysterious" Echo Plus bit copier is now out (or just about to be released). There is a short summary of it in the New Products section on page 89 of the January 1986 issue of *inCider*. It says that "Echo Plus" copies any disk, even the best-protected for your backup." No parameters are needed. "The copy process may take 20 minutes but [heavily-protected] disks can be easily copied." It costs $59.95 from:

Agranat Systems
10 Winthrop Circle
Weston, MA 02193
(518) 266-8718

Also, I tried to work my way through the Karateka softkey in COMPUTIST No. 23, but it didn't seem to work. It was getting caught in step 17 (I think) where the "A851G" command was supposed to reconnect DOS, but didn't for some reason. To get around this, I just booted a system master disk & saved what had to be saved and then looped back to step 15, which seemed to work fine.

Sam Wong
San Diego, CA

GATO Again?

It appears as though the protection scheme for GATO, the WWII submarine program has been changed.

When a friend and I were called upon to back up this program, we naturally checked for a softkey in COMPUTIST. Clay Harrell's softkey, which appeared in COMPUTIST No. 23 was an excellent starting point, but it became clear that the publishers had moved the nibble check routine to elsewhere on the disk.

We ended up scanning the disk with Watson and found the nibble count sequence ($AD $E9 $C0 $A9) had been moved from track $15, sector $0C to track $15, sector $0A. Surprisingly enough, the sequence was in the same place (byte $8C to $8F inclusive) in this new version.

All that is required to deprotect this new version of GATO, is to change the sequence $AD $E9 $C0 $A9 to $A9 $01 $D0 $67 respectively, on a backup disk made with COPYA or equivalent.

Stephen Brown
Willodale, ON
CANADA

Mr. Brown: *I am sure other readers will find this information useful.*

COMPUTIST has had reports of yet another version of GATO in which the disk cannot be copied with COPYA due to altered address marks on every-other track. We hope to present a softkey for this version soon.

Most Wanted List

Need help backing-up a particularly stubborn program?

Send us the name of the program and its manufacturer and we'll add it to our Most Wanted List, a column (updated each issue) which helps to keep COMPUTIST readers informed of the programs for which softkeys are MOST needed. Send your requests to:

COMPUTIST
Wanted List
PO Box 110846-K
Tacoma, WA 98411

If you know how to deprotect unlock, or modify any of the programs below, let us know. You'll be helping your fellow COMPUTIST readers and earning MONEY at the same time. Send the information to us in article form on a DOS 3.3 diskette.

Mouse Calc Apple Computer
Apple Business Graphics Apple Computer
Jane Arktronics
Visiblend Microlab
Catalyst Quark, Inc.
Gutenburg Jr. & Sr. Micromation LTD
Prime Plotter Primesoft Corp.
The Handlers Silicon Valley Systems
The Apple's Core: Parts 1-3 The Professor
Fun Bunch Unicorn
Willy Byte ... Data Trek
Terrapin Logo V2.00 Terrapin Software
Conan Datasoft
Cycloid Sirius Software
Crisis Mountain Synergistic Software
Adventure Microsoft
Olympic Decathlon Microsoft
Cranston Manor Sierra On-Line
Snoggle Broderbund
Robot War Muse
ABM Muse
Mychess II Datamost
E-Z Learner Silicon Valley Systems
Story Tree Scholastic
Agent U.S.A. Scholastic
Snack Attack Datamost
Instant Pascal ALS
The Hobbit Addison Wesley
The Halley Project Mindscape
Captain Goodnight... Broderbund
Handicapping System Sports Judge
Dollars and Sense Monogram

readers' softkey & copy exchange

Jean-Michel Georges's softkey for...

King's Quest

Sierra On-Line
P.O. Box 485
Coarsegold, CA 93614

Requirements:
Apple //c or //e with extended 80-column card
A sector editor
COPYA
Three blank disk sides

When you try to copy the original with something like COPYA, there seems to be no problem. However, when you boot the copy, it doesn't work. Oh, no! A nibble count!

With a disk searcher, I looked for the bytes $8C C0, since a disk read is usually done with a LDA $C08C,X (X being the slot number * 16), and found some strange code on track $E, sector $F. This is their protection!

A little detective work indicates that this sector is loaded into memory at $FF00. This is a strategic location and very difficult to inspect.

The Softkey

The most difficult part was finding where this sector was loaded into memory. From there it is very easy to deprotect the disk. With C.I.A., I searched for the sequence 20 00 FF (JSR $FF00) and found it in four locations. I replaced all of them with EA EA EA (NOP's) and it worked!

1) Copy the three sides of King's Quest with COPYA or similar.

2) Start up your favorite sector editor and make these changes on only the boot side of King's Quest:

(At all four locations, change the bytes "20 00 FF" to "EA EA EA").

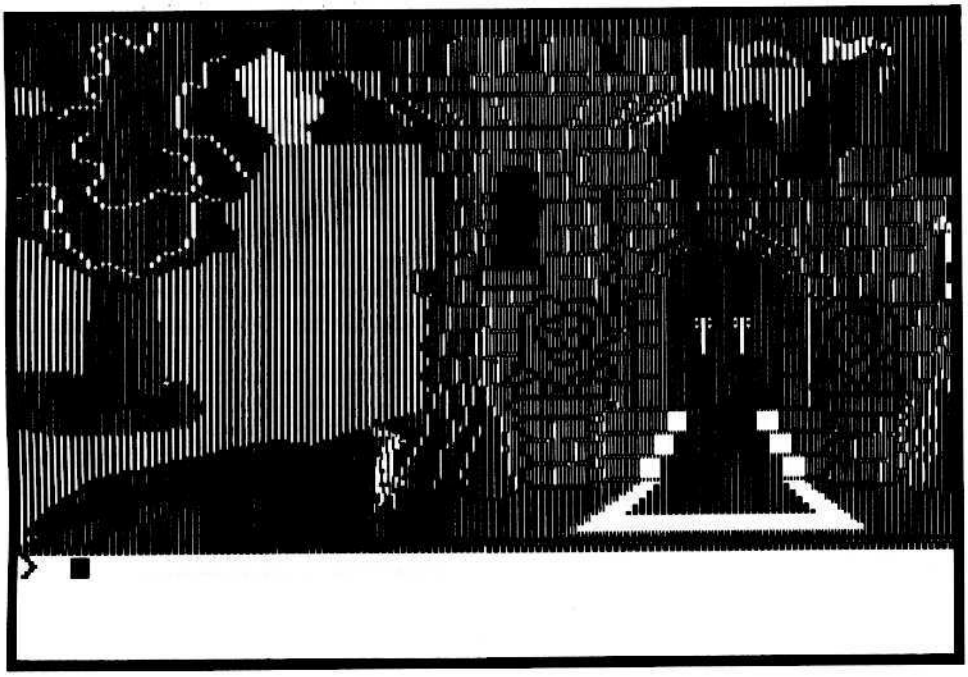
Track	Sector	1st Byte
$0B	$02	$B9
$0B	$0B	$ED
$0B	$0D	$B6
$0C	$01	$48

You now have a COPYAable King's Quest!

Danny Pollak's softkey for...

Mastering the SAT

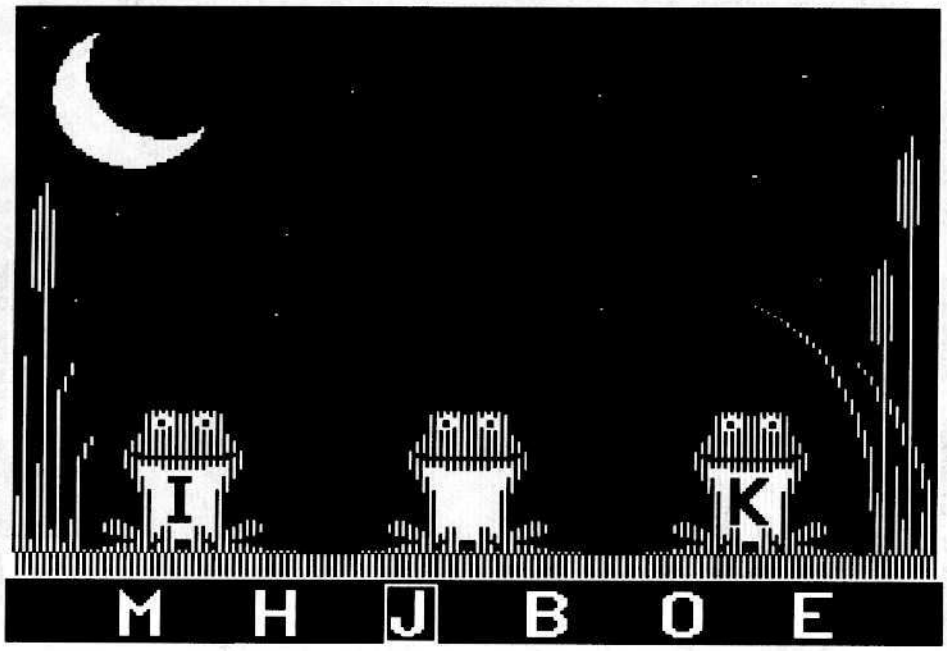
CBS Software
One Fawcett Place
Greenwich, CT 06836
$150.00



Requirements:
Apple]|[Plus or equivalent
Super IOB v1.5
Four blank disks

Mastering The SAT is one of the best preparatory programs of its type. It comes with a manual and four disks. The disks contain a pretest, a posttest, and a series of exercises called "Computer Skill Builders." The manual contains two full length simulated SAT exams, complete with answers. It also contains a section on test taking strategies.

readers' softkey & copy exchange



The Protection

The method used to protect the disks is fairly simple. First of all, an absolutely normal DOS is used. Secondly, tracks $03 and $04 have been written in a strange format, although there is no nibble count. This second item stops a whole disk copier such as COPYA from working. Finally, The Volume Table of Contents (VTOC) has been moved from track $11 sector $00 to track $15 sector $00. This last change keeps a file by file copier, such as FID or Copy][+, from transferring the files to another disk. Obviously, CBS software spent more money developing the program than protecting it.

The Deprotection

Use the following steps to deprotect the four program disks.

1) Boot a regular DOS (preferably a fast one) into memory. Modify it to look for the VTOC at track $15 and initialize a disk with a hello file of (what else?) HELLO. Do this for all four blank disks.

CALL-151
AC01:15
INIT HELLO

2) Reboot with normal DOS. Load Super IOB v1.5 and install the controller located at the end of this article.

3) Follow the prompts and answer no when asked if you wish to initialize the duplicate diskette. The controller copies track $05-$22.

4) When Super IOB is finished, you will have a completely COPYAable disk. Use the same procedure on the other disks and there you have it: a completely deprotected version of CBS' "Mastering The SAT."

controller

```
1000 REM MASTERING THE SAT
1010 TK=5 :LT=35 :ST=15 :LS=15 :CD=WR :FAST
     =1
1020 GOSUB 490 : GOSUB 610
1030 GOSUB 490 : GOSUB 610 : IF PEEK (TRK ) = LT
     THEN 1050
1040 TK=PEEK (TRK ) :ST=PEEK (SCT ) : GOTO 1020
1050 HOME : PRINT "COPYDONE" : END
```

controller checksums

```
1000  - $356B     1030  - $8B51
1010  - $2745     1040  - $A4C5
1020  - $0DC5     1050  - $56A2
```

B. Croome's softkey for...

Easy as ABC

Springboard Software Inc.
7807 Creekridge Cr.
Minneapolis, MN 55435

Requirements:
Super IOB v1.5
A blank disk
Easy as ABC disk

Easy as ABC, according to a friend of mine, was an enjoyable experience for his child, but watching the child's rough handling of the original disk would give him nightmares.

Unfortunately, he was unable to copy it to allow his son to use a copy rather than the original. Fortunately, on examination of the disk I found what appeared to be a near normal disk with an alteration of the epilog bytes for both data and address markers as the protection scheme. The address epilog bytes were changed to AD BB and the data epilog to ED BB. I started by writing a controller for Super IOB that would alter these bytes during the READ portion of the program and normalize it during the WRITE portion. This should have worked but did not as there were all kinds of grinding noises when IOB tried to read some sectors.

Becauase the RWTS looked so normal I decided to use try the Super IOB fast swap controller and use Easy as ABC's own RWTS to read the original and swap in a normal RWTS to normalize the write. This worked admirably.

So, here are the cookbook instructions for "Easy as ABC."

1) Initialize a blank disk with a hello program named "MENU."

INIT MENU

2) Boot up the original disk and when the Applesoft prompt appears, press [ESC][C]. Enter the monitor by typing:

CALL -151

Alternate step 2: *If you've just spent a large quantity of money on one of those fancy cards that can interrupt the program at a moment's notice (Wildcard, Know-Drive, etc.), here's your chance. Wait until the title page appears then press the appropriate switch. Gives you a feeling of power, doesn't it?*

3) Move the protected RWTS to a safe spot.

1900<B800.BFFFM

4) Boot a 48K slave disk with no hello program.

C600G

5) Save the original RWTS to the same disk as your Super IOB program.

BSAVE ABC RWTS,A$1900,L$800

6) Type in the fast swap contoller given below and run Super IOB v1.5.

That's all there is to it. As an added bonus you can add a fast DOS like Pronto-Dos to speed up the boot and disk access time.

controller

```
1000 REM ABC CONTROLLER
1010 TK=3 :ST=15 :LT=35 :LS=15 :CD=WR :FAST
     =1
```

readers' softkey & copy exchange

```
1020 GOSUB 360 : GOSUB 490 : GOSUB 610
1030 GOSUB 360 : GOSUB 490 : GOSUB 610 : IF PEEK
     (TRK ) = LT THEN 1050
1040 TK = PEEK (TRK ) :ST = PEEK (SCT ) : GOTO 1020
1050 HOME : PRINT "COPY^ DONE" : END
10010 PRINT CHR$ (4 ) "BLOAD^ RWTS.ABC,A$1900"
```

controller checksums

```
1000  - $356B      1040  - $8FE3
1010  - $BC8F      1050  - $1516
1020  - $51C2      10010 - $2562
1030  - $6356
```

Jordi Le Vant's softkey for...

Space Shuttle

Activision

Requirements:
COPYA
A sector editor
A blank disk

The only protected tracks on the Space Shuttle disk are tracks 4, 5 and 6 which have modified address trailers. To start, we'll tell DOS not to read the epilogues.

1) First boot a DOS 3.3 disk, then enter the monitor and make the following changes.

CALL-151
B99C:EA EA
3D0G

2) RUN COPYA and copy the Space Shuttle disk.

After the main loader for the program is loaded, it makes a small modification to RWTS before reading the three abnormal tracks. These three tracks aren't normally copyable because the second byte in the address field trailer is changed, and they are also different for each protected track. The modification made to RWTS changes it so instead of checking to make sure the trailer is a specific byte (normally $AA), it simply reads the byte and saves it, then stores it at locations $00, $01, $02 for tracks 4, 5 and 6 respectively. So after reading these three tracks, these locations should have the following values.

Address	Value
$0000	$AB
$0001	$BD
$0002	$DE

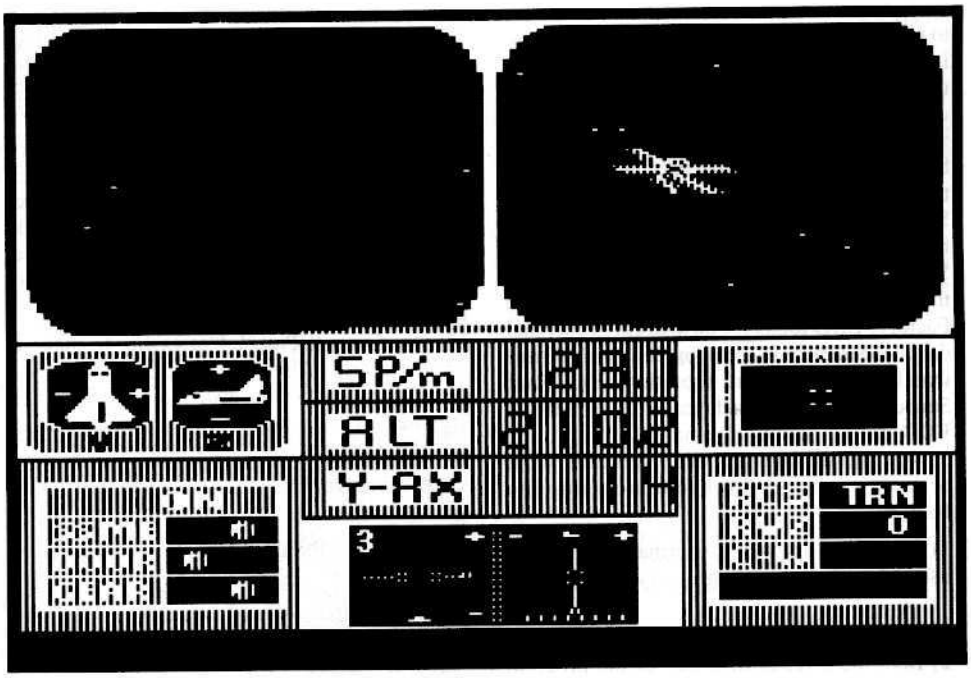
Because everything that's loaded into memory is encoded, the loader decodes the data after loading it using these three values. Since these values are no longer on the disk, we need to put them at these locations before it jumps to the decode routine.

3) Get out your sector editor and make the following changes to **track 0, sector $0B** of the copy. (This is a JSR to the routine we'll add in step 4.)

Byte	From	To
$48	$A0	$20
$49	$00	$E0
$4A	$98	$41

4) Add the following bytes from byte $E0 to $EF on the same track and sector. This routine stores the required values in $00-$02.

```
Byte    Values
-----------------------------------------
xxE0:   A9 AB 85 00 A9 BD 85 01
xxE8:   A9 DE 85 02 A0 00 98 60
```



readers' softkey & copy exchange

While you're at it, if you want to shorten the time the the title page stays on, change track 0, sector 10 ($0A), byte $2C, from $28 to $01 for the shortest time.

And that's all, folks.

Doug Walker's softkey for...

The Factory

Sunburst Education
39 Washington Ave.
Pleasantville, NY 10570

Requirements:
At least 48K
A way into the monitor
Super IOB with swap controller
A blank disk

The Factory is a neat learning tool. It teaches planning and logic to those using the program. The object is to build an assembly line which will produce the item specified by the computer.

Sunburst Communications develops some interesting Software packages, with some equally interesting protections as well. If any of you have ever tried to back up any of the Sunburst series with a nibble copier, I'm sure you've met the same frustration as I. Sunburst uses an altered RWTS to read their disks. Their routines are designed to read altered address headers as well as altered data headers. Each address header and data header is normal until the last byte (i.e. D5 AA xx). The correct bytes to place at the end of each header can be found in the write translate table at $BA29. This is the table used by the prenibble routine to convert a byte to its encoded form that will be written to the disk. The address header's third byte is found by indexing into the table using the track number (the location of this byte is $BA29 + (trk#)). The same is done for the data header using instead the numbers starting at $BA34. These altered headers are the reason that bit copiers like EDD III can't copy The Factory. EDD looks for an address header and data header early in the copy process and uses that header for the rest of the disk. Since Sunburst is changing each header on each track, EDD can't find the information.

Now that the background information is out of the way, let's get down to deprotecting The Factory.

1) Initialize a disk with a normal or fast DOS using the filename "Logo".

INIT LOGO

2) Boot the original disk and let the program load until you see the Sunburst logo appear. Drop into the monitor.

3) Move The Factory's RWTS to a safe place.

1900<B800.BFFFM

4) Insert a slave disk and reboot.

C600G

5) Save the RWTS to your Super IOB disk.

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

6) Install a Swap Controller into Super IOB and set it to load "RWTS.FACTORY" (or whatever you called it) and copy tracks $3-$22 of the disk.

The Factory has now been completely deprotected and is COPYAable.

Thomas J. Scott's softkey for...

Visidex 1.1E

Paladin
2895 Zanker Rd.
San Jose, CA 95134

Requirements:
A good copy program (that will copy DOS)
A way into the monitor

Visidex is a filing and schedule program that makes use of a clock card. The features are excellent, like the data retrieval just to mention one. It is a single load program so I thought it would be an excellent candidate to put on a ROM chip (to use with the "Quick Loader" from SRCG). So off I went to my back issues of COMPUTIST for a softkey. Sure enough I found one; but, alas, it did not work! I managed to find two other softkeys, but they did not work either! Well, it seems that they were for a different version. For any of you out there who may have version 1.1E, here is a softkey.

1) Use Copy][Plus 4.4 or earlier to make a working copy. (Use regular copy, not bit copy.) EDD III, Locksmith 4.0, Copy][Plus 5.2 regular or bit copy would not produce a working copy.

2) Format two blank disks. Call one "disk #1" and the other one "disk #2."

3) Copy only the DOS from the working copy to disk #1.

4) Boot up the working copy and when it's up and running, pop into the monitor by your favorite method.

5) Re-enter DOS by typing

9D84G

6) Remove the working copy and insert disk #2.

BSAVE VISIDEX1,A$803,L$4500
BSAVE VISIDEX2,A$6000,L$100

7) Remove disk #2, insert disk #1, and reboot.

PR#6

8) After the drive stops and FILE NOT FOUND appears (this gets the Visidex DOS into memory and enables us to perform the next step) remove disk #1 and insert disk #2. Type:

BLOAD VISIDEX2

This is the program that will BLOAD and run the Visidex program itself.

9) Remove disk #2 and insert disk #1.

INIT VISIDEX

10) After the drive stops type

DELETE VISIDEX

11) Remove disk #1 and insert disk #2.

BLOAD VISIDEX2

This is the main Visidex file.

12) Remove disk #2 and insert disk #1.

BSAVE VISIDEX,A$803,L$4500

Done!

Ed Croft's softkey for...

Sherlock Holms

Bantam Software
666 Fifth Avenue
New York, NY 10103
$39.95

Requirements:
64K Apple][+, //e, //c
COPYA
A sector editor
A blank disk

Sherlock Holmes from Bantam Software is a hi-res adventure game. It comes with playing instructions, a ship map and a 32 page booklet containing the first three chapters of the adventure/story you are playing. The hires pictures take up about a fourth of the screen and

are beautifully done. Unfortunately the program relies on knowledge of the ship's map to move around. (i.e. No "GO NORTH"). Specific locations on the map must be referred to. Play is also slow due to disk access after every entry. This plus the fact that the disk is double sided makes it a prime canidate for deprotection.

The protection scheme used on Sherlock Holmes does not prevent it from being copied by COPYA. But the program then checks for byte sequences on track $22. We need to defeat that, of course.

A cookbook procedure:

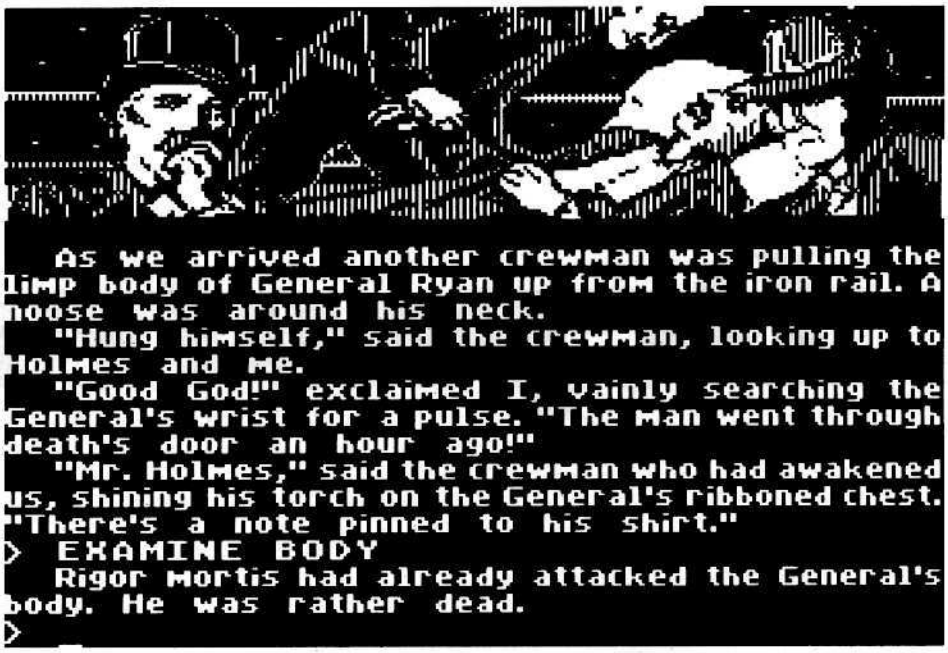
1) Boot normal DOS 3.3.

2) Copy both sides of the disk with COPYA. Ignore the UNABLE TO READ error on the front side, because we got all we needed already.

3) Start up your sector editor (I use Tricky Dick) and make the following changes:

track	sector	byte	from	to
$16	$0	$32	$4C	$EA
$16	$0	$33	$32	$EA
$16	$0	$34	$D0	$EA

Your copy is now deprotected. (Elementary, my dear Watson!!)



Charles Taylor's softkey for...

The Bard's Tale

Electronic Arts
2755 Campus Dr.
San Mateo, CA 94403

Requirements:
64K (for the game)
Three blank disk sides
A sector editor
A copy program that can ignore errors or skip tracks

The protection on The Bard's Tale (a nice adventure) is similar to, if not exactly the same as, the protection used on Seven Cities of Gold (another EA release). For reference, see the Electronic Arts softkey in COMPUTIST No. 24, page 10. This is a totally normal DOS 3.3 sector format with the exception of track six on the boot disk.

1) Start by copying all three sides onto your blank disks using something like Copy][Plus or Locksmith Fast Copy (or even Super IOB if you're clever!).

2) Get out your sector editor and read in track 1, sector $E of the boot disk. Change bytes $47-$49 from 20 F8 A0 to 18 60 40. Write the sector back out.

Now read in track 1, sector $B and change bytes $47-49 the same way. Write the sector back out.

3) Put away all your diskbusting toys and play your backup copy of the game.

If necessary, try different combinations of sector edits mentioned in the Electronic Arts article in COMPUTIST No. 24.

Deprotecting

by William Clarke

Millionaire
Blue Chip Software
19824 Ventura Blvd.
Suite 125
Woodland Hills, CA 91364
$79.95

Requirements
Apple //c or //e with extended 80 column card
DOS System Master
Two disks intialized with DOS 3.3

The idea of investing in the stock market has always intrigued me. I guess it's because television and movies portray the stock market as a place where lots of money can be made. The only problem is that in real life it's a place where lots of money can be lost too! I'd love to dabble in the market and make a lot of money, but the thought of losing my hard earned bucks has been holding me back. What am I to do?

Blue Chip Software has a partial solution for people like me. Millionaire is a stock market simulation game which allows me to play the stock market all I want without the risk of losing all my money. Of course, I don't make any money either! The object of the game is to make a million dollars from your $10,000 venture capital. Your job is to analyze the current economic climate and "wisely" buy and sell stocks and options from among 15 corporations. Millionaire provides you with weekly information concerning the status of each of these companies. You decide when the time is right to buy, sell, borrow or simply wait.

I was sorry to see that Millionaire is copy protected, so I set out to remedy that situation. I also wanted to try using the RAMcard method as described by Ken Greenlaw in COMPUTIST No. 16. I am happy to report that Millionaire can be completely deprotected (COPYAable) to the extent that you can even list the programs (written in BASIC) and make your own modifications.

Background

As discussed by Ken Greenlaw, the trick to using the extra 64K random access memory (RAM) card to deprotect programs is to have the program loaded into the RAM card when the program disk is booted. Once the program is running in the RAM card's memory (also known as "auxiliary memory") you can regain control over your Apple by simply pressing the Reset key. This switches the Apple back to using its main memory with the protected program sitting in auxiliary memory. Once you have control, you can examine what's in the auxiliary memory, i.e. your protected program.

So, how do you get your protected program to boot into the auxiliary memory? Apple has provided a short program in Read Only Memory (ROM) that we will use. This subroutine is called "XFER", short for "TransFER", and its function is to transfer control of the computer from a program in main memory to a program in auxiliary memory or vice versa. We need to tell XFER that we want to transfer control of the computer to a program in auxiliary memory. The program we need is one that will boot the copy protected disk. This program also already exists in the ROM of the disk controller card at $C600.

Therefore we must tell XFER two things:

1) The location of the program in auxiliary memory which will receive control ($C600).

2) The direction of the transfer (main to auxiliary).

XFER expects to find the location of the program to receive control at $3ED and $3EE in low-high format. We want to put $00 into $3ED and $C6 into $3EE. Then XFER needs to know in which memory (main or auxiliary) the program will be executing. Here things get a little more complicated because we can use different combinations of main and auxiliary memory. Locations $0000 through $01FF and $0200 through $BFFF of main and auxiliary memory can be used independently of each other. The carry bit of the Processor Status register signals the direction of the transfer and the overflow bit selects which block of $0000-01FF (main or auxiliary) will be used.

Millionaire

We want to use all of auxiliary memory ($0000-BFFF) since all of main memory would be used if we were booting the disk normally. Therefore, we must set (place a "1" in) both the carry and overflow bits.

Booting Millionaire Into Auxillary Memory

The program "XFER.BOOT" written by Ken Greenlaw does all this for us. I typed in XFER.BOOT and saved it on a normal DOS 3.3 diskette. Then I placed the Millionaire disk into drive 1, typed "PR#3" to turn on the 80-column card and executed XFER.BOOT by typing CALL 768. My disk drive started whirring and I saw the introductory message Millionaire displays when booted. Next I pressed RESET and I was faced with a blank screen with only the APPLESOFT prompt. Well, if Ken was right, Millionaire should be waiting for me in auxiliary memory.

Here are the steps to boot Millionaire into auxiliary memory:

1) Load your copy of XFER BOOT into memory.

BLOAD XFER.BOOT

2) Turn on your extended 80 column card.

PR#3

3) Put Millionaire disk into drive #1.

4) Start up the XFER BOOT program.

CALL 768

5) Return to Applesoft by pressing RESET.

Strategy #1

My strategy to deprotect Millionaire was to first examine the DOS Millionaire used. To do this I needed to transfer Millionaire's DOS (M.DOS) from auxiliary to main memory. This task was easily accomplished with the use of a second subroutine called AUXMOVE in Apple's ROM, at location $C311. When called, AUXMOVE takes a block of memory and moves it from main memory to auxiliary memory or vice versa. You need to supply AUXMOVE with the addresses for the beginning (in locations $3C and $3D) and end ($3E,$3F) of the block of memory to be moved and the destination ($42,$43) of the block. Also, the direction of the memory move is indicated by the status of the carry bit in the Processor Status Register; if the carry bit is set, the move will be from main to auxiliary memory and vice versa if the carry bit is clear.

To move M.DOS from auxiliary to main memory, I used a modification of the program RESTORE written by Ken. MOVE, as listed below, sets up AUXMOVE to move a block of memory from main to auxiliary or from auxiliary to main. As with RESTORE, MOVE is called via the CTRL-Y vector. If you are in the Monitor and you type a command with the format $AAAA<$BBBB.$CCCCCTRL-Y<CR>, a subroutine in the Monitor is executed which places $AAAA in location $42 and $43 (in low-high format), $BBBB in $3C and $3D and $CCCC in $3E and $3F. Then execution continues at location $3F8 which usually contains an instruction to jump to a machine language subroutine that you can specify. Unless you change it, the instruction that DOS puts there on boot-up is a jump to $FF65 (4C 65 FF; JMP $FF65). $FF65 is one of the entry points to the Monitor; a jump to $FF65 is almost like a CALL -151. However, you can change the instruction at $3F8 to point to a routine of your choosing. We will place a jump to location $300 at $3F8 (4C 00 03) which is where our MOVE routine is located.

To use MOVE to move memory from auxiliary to main, enter the Monitor by typing CALL -151 and type 3F8:4C 00 03. Now enter the destination, starting location and ending location of the memory to be moved in the form: DEST<START.ENDCTRL-Y. To transfer a block of main memory to auxiliary memory, type 3F8:4C 05 03 and then:

DEST<START.ENDCTRL-Y

```
        * * * MOVE * * *
300 CLC     MOVE AUX TO MAIN
301 JSR $C311
304 RTS
305 SEC     MOVE MAIN TO AUX
306 JSR $C311
309 RTS
```

You can enter this program directly from the Monitor as follows:

CALL -151
300:18 20 11 C3 60 38 20 11
308:C3 60

You can save this to a DOS 3.3 disk by typing BSAVE MOVE, A$300, L$0A. After loading MOVE, I moved M.DOS from auxiliary to main memory with the following commands:

1) Load the MOVE program.

BLOAD MOVE

2) Enter the Monitor.

CALL -151

3) Set up the Y vector.

3F8:4C 00 03

4) Initiate the move.

9600<9600.BFFFY

This sequence of instructions placed what was in locations $9600 through $BFFF of auxiliary memory (hopefully M.DOS!) into main memory beginning at $9600 and extending through $BFFF.

When I examined the memory between $9600 and $BFFF, I found that valid looking code was now present. (If you are interested, you can examine M.DOS in detail to learn about some of the protection schemes used by Blue Chip Software. For example, look at the address fields read by RDADDR [$B944-B99F]). If this was M.DOS then I should be able to access the Millionaire disk with it; i.e. CATALOG the disk or possibly even INITialize a disk with it.

I crossed my fingers and typed CATALOG. Alas, all that happened was a beep from the speaker accompanied by an error message. No need to worry yet, its possible that as part of the protection plan the programmers at Blue Chip changed the CATALOG command in the DOS command table. I was still in the Monitor, so I typed A56EG to access the CATALOG command handler directly and crossed my fingers again. Lo and Behold, the disk drive began to whirr and a catalog appeared on the screen! The following is the catalog listing of Millionaire:

```
DISK VOLUME  254

 A  006  HELLO
 A  036  INITIAL
 A  092  PLAY
 B  003  CHAIN
 T  048  SAVE
 T  018  INDUST
 T  035  STOCKS
 T  010  COMMON
 T  078  MESDATA
 T  041  DESCRIP
 T  004  PLAYER
```

Strategy #2

Since I was able to use M.DOS to CATALOG the Millionaire disk, I thought I could use the same procedure to read each of Millionaire's files. Then I could use normal DOS 3.3 to write the files to a normal DOS 3.3 disk and thereby have an unprotected version of Millionaire!

My planned procedure was:

1) Load DOS 3.3 into main memory by booting a normal DOS 3.3 disk.

2) Load Millionaire into auxiliary memory by using XFER.BOOT.

3) Move DOS 3.3 to auxiliary memory (at $4600, below M.DOS) by using MOVE.

At this point, auxiliary memory will contain a copy of both M.DOS starting at $9600 and DOS 3.3 at $4600.

4) Move M.DOS into main memory with MOVE (at $9600 which will overwrite DOS 3.3).

5) Use M.DOS to read in a file from the protected disk. (See below)

6) Move DOS 3.3 from auxiliary memory to main memory with MOVE (at $9600 which will overwrite M.DOS).

7) Use DOS 3.3 to write the Millionaire file to a DOS 3.3 disk. (See below)

8) Go to step 4 until done with all the files!

For the three Applesoft files, this procedure was really easy. For the TEXT files, the process wasn't so easy, but certainly not impossible. The binary file, CHAIN, is on your DOS 3.3 System Master and can be transferred to your unprotected Millionaire disk with FID.

The Nitty Gritty

Well, I already had a copy of M.DOS in auxiliary memory (at $9600), now I needed to place a copy of DOS 3.3 in auxiliary memory at $4600. I booted a normal DOS 3.3 disk and loaded MOVE into memory. Next, I set up the Y vector to move memory from main to auxiliary memory by entering the Monitor and typing 3F8:4C 05 03. Then I moved DOS to $4600 in auxiliary memory by typing 4600<9600.BFFFY. Now with both M.DOS and DOS 3.3 in auxiliary memory, I was ready to begin!

To move DOS 3.3 to auxiliary memory:

1) Boot a DOS 3.3 disk (without turning off your computer!!)

2) BLOAD MOVE

3) CALL -151

4) 3F8:4C 05 03

5) 4600<9600.BFFFY

I started with the Applesoft programs. I transferred M.DOS to main memory as I described earlier. With M.DOS in main memory, I placed the Millionaire disk into drive #1 and loaded Millionaire's HELLO program into main memory:

LOAD HELLO

After the disk drive stopped, I checked to see if the program was loaded successfully by typing C (to return to Applesoft BASIC from the Monitor) and then LIST. Sure enough, a short program which loads the program INITIAL was present in memory. My next step was to replace M.DOS in main memory with DOS 3.3 and save the HELLO program on a disk previously initialized with DOS 3.3.

1) Enter the Monitor.

CALL -151

2) Set up the Y vector to move auxillary memory to main memory

3F8:4C 00 03

3) Move DOS 3.3 to main memory.

9600<4600.6FFFY

4) Place DOS 3.3 disk in drive and save the program.

SAVE HELLO

All was proceeding well and according to plan, so I moved M.DOS from auxiliary memory to main memory (9600<9600.BFFFY). After placing the Millionaire disk in the drive, I loaded the next program (LOAD INITIAL) and repeated the above procedure to save INITIAL to my DOS 3.3 disk. Finally I transferred the last Applesoft program, PLAY, to my DOS 3.3 disk.

So far, so good. However, the Text files, which contain the data for the game, were going to be another story.

To transfer the Text files, I had to do some snooping through the Applesoft programs INITIAL and PLAY. I studied the sections of the programs that read in the data from the Text files. I discovered that 5 of the 6 Text files were random access Text files because the L (record length) parameter was used when opening them. The 6th file was a sequential Text file called SAVE which is used to save data from an incomplete game and did not need to be transferred as it would be created when a game was saved.

Random access files are rather difficult to deal with since they are composed of records, each of which has a constant length (# bytes as specified by the length parameter) but the records can contain different numbers of fields of varying length. See the DOS manual (pg. 81) for more information about random access files. Therefore, I had to write separate programs to read in the records from each file with M.DOS and write the records out to a DOS 3.3 file.

Each of these programs consists of two parts. The first section (lines 10 through 300) is used with M.DOS to read the Text files from the protected disk. The second part (lines 500 through 800) is used with DOS 3.3 to write the files to your unprotected disk. Type in these programs and save them on a DOS 3.3 diskette. The procedure to use each of these programs is described below.

The only unusual aspect of this procedure is in step 8. At this point you have read the file from the protected disk and have DOS 3.3 in memory. You are just about to execute the second half of the program to write the file to your DOS 3.3 disk with a GOTO 500 command.

If you try to continue execution of the program with GOTO 500, DOS will tell you that this is "NOT A DIRECT COMMAND". The reason for this is that DOS 3.3 does not realize that a program has been RUN.

Remember, we executed the program when M.DOS was in memory. To get around this problem, we need to circumvent the DOS routine that checks to see if a BASIC program is running. This is the purpose of step #8. When you type GOTO 500, a subroutine in DOS will check to see if a BASIC program has been RUN and will return with a negative finding. Changing location $A677 from $38 to $18 forces that subroutine to return with an affirmative signal.

Procedure to Transfer Text Files

1) With DOS 3.3 in main memory, LOAD a the appropriate text file reader/writer program into memory.

2) Enter the MONITOR (CALL -151) and transfer M.DOS from auxiliary memory to main memory.

3) Place the protected Millionaire disk into the drive.

4) Type [CTRL-C] to return to Applesoft BASIC.

5) Type RUN to run the text file program to read the data.

When the program stops (at line 300) the data will be in memory.

6) Enter the MONITOR (CALL -151) and replace M.DOS with DOS 3.3.

7) Place your DOS 3.3 disk into the drive.

8) Type A677:18 to tell DOS that a BASIC program is running.

9) Type GOTO 500 to execute the second half of the program which writes the TEXT files to disk.

10) Go to step 1 until all files have been read.

Now you have a deprotected disk of Millionaire that is COPYAable! Don't forget to transfer the CHAIN program from your DOS SYSTEM MASTER with FID. Place the disk in your drive and type RUN INITIAL and you'll begin.

Have fun with your unprotected version of Millionaire. Now you can develop some Advanced Playing Techniques (APT's) since you can load and list both BASIC programs the game uses. Enjoy!

Text File Read/Write Programs

Indust

```
10 D$ = CHR$ (13 ) + CHR$ (4 )
20 DIM A$(10 ,91 ) : DIM L(10 ,20 ) : DIM A1$(10
     ,20 )
30 PRINT D$ "OPEN^ INDUST,L400"
40 FOR I = 1 TO 10
45 N = 0
50 PRINT D$ "READ^ INDUST,R" I
60 FOR J = 1 TO 91
70 INPUT A$
80 IF VAL (A$ ) = 0 THEN N = N + 1 : L(I ,N ) = J : A1$(I
     ,N ) = A$ : GOTO 70
90 A$(I ,J ) = A$
100 NEXT J
110 NEXT I
120 PRINT D$ "CLOSE^ INDUST"
300 END
500 PRINT D$ "OPEN^ INDUST,L400"
510 FOR I = 1 TO 10
515 N = 1
520 PRINT D$ "WRITE^ INDUST,R" I
530 FOR J = 1 TO 91
535 IF L(I ,N ) = J THEN PRINT A1$(I ,N ) : N = N + 1
540 PRINT A$(I ,J )
550 NEXT J
560 NEXT I
570 PRINT D$ "CLOSE^ INDUST"
800 END
```

Common

```
10 D$ = CHR$ (13 ) + CHR$ (4 )
20 DIM A$(5 ,91 )
30 PRINT D$ "OPEN^ COMMON,L400"
40 FOR I = 1 TO 5
50 PRINT D$ "READ^ COMMON,R" I
60 FOR J = 1 TO 91
70 INPUT A$(I ,J )
80 NEXT J
90 NEXT I
100 PRINT D$ "CLOSE^ COMMON"
300 END
500 PRINT D$ "OPEN^ COMMON,L400"
510 FOR I = 1 TO 5
520 PRINT D$ "WRITE^ COMMON,R" I
530 FOR J = 1 TO 91
540 PRINT A$(I ,J )
550 NEXT J
560 NEXT I
570 PRINT D$ "CLOSE^ COMMON"
800 END
```

Stocks

```
10 D$ = CHR$ (13 ) + CHR$ (4 )
20 DIM A$(21 ,91 ) : DIM L(21 ,20 ) : DIM A1$(21
     ,20 )
30 PRINT D$ "OPEN^ STOCKS,L400"
40 FOR I = 1 TO 21
45 N = 0
50 PRINT D$ "READ^ STOCKS,R" I
60 FOR J = 1 TO 91
70 INPUT A$
80 IF VAL (A$ ) = 0 THEN N = N + 1 : L(I ,N ) = J : A1$(I
     ,N ) = A$ : GOTO 70
90 A$(I ,J ) = A$
100 NEXT J
110 NEXT I
120 PRINT D$ "CLOSE^ STOCKS"
300 END
500 PRINT D$ "OPEN^ STOCKS,L400"
510 FOR I = 1 TO 21
515 N = 1
520 PRINT D$ "WRITE^ STOCKS,R" I
530 FOR J = 1 TO 91
535 IF L(I ,N ) = J THEN PRINT A1$(I ,N ) : N = N + 1
540 PRINT A$(I ,J )
550 NEXT J
560 NEXT I
570 PRINT D$ "CLOSE^ STOCKS"
800 END
```

Mesdata

```
10 D$ = CHR$ (13 ) + CHR$ (4 )
20 DIM A$(300 )
30 PRINT D$ "OPEN^ MESDATA,L65"
40 FOR I = 1 TO 300
50 PRINT D$ "READ^ MESDATA,R" I
70 INPUT A$(I )
90 NEXT I
100 PRINT D$ "CLOSE^ MESDATA"
300 END
500 PRINT D$ "OPEN^ MESDATA,L65"
510 FOR I = 1 TO 300
520 PRINT D$ "WRITE^ MESDATA,R" I
540 PRINT A$(I )
560 NEXT I
570 PRINT D$ "CLOSE^ MESDATA"
800 END
```

Descrip

```
10 D$ = CHR$ (13 ) + CHR$ (4 )
20 DIM M$(15 ) : DIM W$(15 ,14 ) : DIM W1$(15 ,14
     ) : DIM W2$(15 ,14 )
25 DIM W3$(15 ,14 )
30 PRINT D$ "OPEN^ DESCRIP,L650"
40 FOR I = 1 TO 15
50 PRINT D$ "READ^ DESCRIP,R" I
55 INPUT M$(I )
60 FOR J = 1 TO 14
70 GET W$(I ,J )
80 ON VAL (W$(I ,J ) ) GOTO 90 ,100 ,110
90 INPUT W1$(I ,J ) : GOTO 120
100 INPUT W1$(I ,J ) ,W2$(I ,J ) : GOTO 120
110 INPUT W1$(I ,J ) ,W2$(I ,J ) ,W3$(I ,J )
120 NEXT J
130 NEXT I
140 PRINT D$ "CLOSE^ DESCRIP"
300 END
500 PRINT D$ "OPEN^ DESCRIP,L650"
510 FOR I = 1 TO 15
520 PRINT D$ "WRITE^ DESCRIP,R" I
530 PRINT M$(I )
540 FOR J = 1 TO 14
550 PRINT W$(I ,J );
560 ON VAL W$(I ,J ) GOTO 570 ,580 ,590
570 PRINT D$ "CLOSE^ DESCRIP"
580 PRINT W1$(I ,J ) : PRINT W2$(I ,J ) : GOTO 600
590 PRINT W1$(I ,J ) : PRINT W2$(I ,J ) : PRINT
     W3$(I ,J )
600 NEXT J
610 NEXT I
800 END
```

Player

```
10 D$ = CHR$ (13 ) + CHR$ (4 )
20 DIM PL$(15 ) : DIM LA(15 ) : DIM HI(15 ) : DIM
     ST$(15 )
30 PRINT D$ "OPEN^ PLAYER,L40"
40 FOR I = 1 TO 15
50 PRINT D$ "READ^ PLAYER,R" I
60 INPUT PL$(I ) ,LA(I ) ,HI(I ) ,ST$(I )
70 NEXT I
80 PRINT D$ "CLOSE^ PLAYER"
300 END
500 PRINT D$ "OPEN^ PLAYER,L40"
510 FOR I = 1 TO 15
520 PRINT D$ "WRITE^ PLAYER,R" I
530 PRINT PL$(I ) : PRINT LA(I ) : PRINT HI(I )
     : PRINT ST$(I )
540 NEXT I
550 PRINT D$ "CLOSE^ PLAYER"
800 END
```

Ultimaker IV, an Ultima IV ...

Character

by Danny Pollak

Requirements:
Apple][Plus or better
Ultima IV

Ulitima IV: Quest Of The Avatar is the newest release in the Ultima series. It is definitely a long-play adventure game, in that you must master eight different virtues in order to win the game. It took a lot of time and energy to think up all the aspects of this game.

Player attributes are stored on the Britainia disk. Like Ultima III and Ultima II, most of the information is stored in binary coded decimal format. One notable difference is that letters are no longer used to represent certain characteristics (ie. health, type and sex). Instead, certain hexadecimal values are used to represent these attributes.

Most of the characteristics that we are going to be dealing with are found in the file named ROST. The value which represents the number of characters currently in the party is contained in the file named PRTY.

Table 1 shows the 31 bytes which make up each character

Table 2 (pg. 18) shows the bytes which represent the possessions that belong to the entire party. The bytes start at byte $108 of the file ROST.

Stones, Runes and etc.

Unlike most of the other locations, the two for stones and runes are significantly different. Each bit in the byte represents a flag for a certain stone or rune. If the bit is on (a one) then the party is holding the particular item that is represented by that bit. A similar technique is used for the two bytes which represent the items and the 3 part key except that only bits 0-2 are used. (see Table three)

Table 1
Character Bytes

0-14	Character Name in ASCII
15	$00 marking end of name
16	Sex ($5C-Male $7B-Female)
17	Type (0-7)
18	Health ($C4=D, $C7=G, $D0=P, $D3=S)
19	Strength (0-99)
20	Dexterity (0-99)
21	Intelligence (0-99)
22	Magic Points (0-99)
23-24	Hit Points (0-9999)
25-26	Maximum Hit Points (0-9999)
27-28	Experience (0-9999)
29	Weapon Currently In Use (0-15)
30	Armour Currently In Use (0-7)

Table 3

Bit	Stone	Rune	Item	3p key
0	Black	Humility	Candle	Courage
1	White	Spirituality	Book	Love
2	Purple	Honor	Bell	Truth
3	Orange	Sacrafice		
4	Green	Justice		
5	Red	Valor		
6	Yellow	Compassion		
7	Blue	Honesty		

Note: The game keeps track of how well you are doing in each of the eight virtues. These can be changed, if you know the right locations. I will leave them up to you to find.

Entering and Using Ultimaker IV

Ultimaker IV consist of a single BASIC program. Type in the BASIC program, being sure to save it before RUNning it.

SAVE ULTIMAKER IV

To use the program, insert a copy of the Britainia disk when prompted and then press any key. A roster of the eight characters on the disk, along with several other options, will be displayed on the screen. Just choose an option and your'e off.

Pressing RETURN without typing anything will leave an option unchanged. Pressing RETURN after typing a number will change the item you are currently editing to the value which you entered. If you type the maximum number of characters, then the number is entered automatically.

Pressing ESC at any time will leave the menu you are in and go to the previous one. If you press ESC while editing a character's attributes or the parties' possesions, you will return to the main menu and any changes you have made will not be entered. Pressing ESC at the main menu will bring you to the INSERT DISK prompt. Pressing ESC at the INSERT DISK prompt will exit the program.

If you are editing the weapon or armour currently in use, you must enter the letter which corresponds to the particular weapon or armour you wish to select. Table 4 contains the armour and weapons and their corresponding letters.

If you are changing the character type, then use table 5 to determine the correct number to select.

Editor

If you are editing the sex of a character, enter M for Male or F for Female. If you are editing a character's health, enter G for Good, D for Dead, P for Poisoned, or a S for Sleeping.

Good luck!

Table 4

Armour	Weapons	
A) None	A) Hands	I) Crossbow
B) Cloth	B) Staff	J) Flaming Oil
C) Leather	C) Dagger	K) Halberd
D) Chain	D) Sling	L) Magic Axe
E) Plate	E) Mace	M) Magic Sword
F) Magic Chain	F) Axe	N) Magic Bow
G) Magic Plate	G) Sword	O) Magic Wand
H) Mystic Robe	H) Bow	P) Mystic Sword

Table 5

Character Type	
Ø) Mage	4) Tinker
1) Bard	5) Paladin
2) Fighter	6) Ranger
3) Druid	7) Sheperd

Ultimaker BASIC Program

```
10 REM <><><><><><><><><><>
20 REM <                    >
30 REM <    ULTIMAKER IV    >
40 REM <                    >
50 REM <  BY DANNY POLLAK   >
60 REM <                    >
70 REM <><><><><><><><><><>
80 REM
90 IF PEEK (104 ) < > 16 THEN POKE 103 ,1 : POKE 104 ,16 : POKE 4096 ,Ø : PRINT CHR$ (4 ) "RUN^ ULTIMAKER^ IV"
100 DIM WE$(15 ) ,WE(15 ) ,MI(25 ) ,MI$(25 ) ,CH(6 )
110 DEF FN B1(X ) = INT ( PEEK (CA + X ) / 16 ) * 10 + PEEK (CA + X ) - INT ( PEEK (CA + X ) / 16 ) * 16
120 DEF FN B2(X ) = INT ( PEEK (CA + X ) / 16 ) * 1000 + ( PEEK (CA + X ) - INT ( PEEK (CA + X ) / 16 ) * 16 ) * 100 + FN B1(X + 1 )
130 FOR X = Ø TO 7 : READ Z$(X ) : NEXT : FOR X = Ø TO 3 : READ H$(X ) : NEXT
140 FOR X = Ø TO 15 : READ WE$(X ) : NEXT : FOR X = Ø TO 7 : READ AR$(X ) : NEXT
150 FOR X = Ø TO 7 : READ RE$(X ) : NEXT : FOR X = Ø TO 25 : READ MI$(X ) : NEXT
160 FOR X = Ø TO 7 : READ S$(X ) : NEXT : FOR X = Ø TO 7 : READ R$(X ) : NEXT
170 FOR X = Ø TO 5 : READ I$(X ) : NEXT : FOR X = Ø TO 6 : READ CH$(X ) : NEXT
180 CH$ = CHR$ (8 ) :CU$ = CHR$ (21 ) :CM$ = CHR$ (13 ) :ESC$ = CHR$ (27 ) :CG$ = CHR$ (7 )
190 D$ = CHR$ (13 ) + CHR$ (4 ) : PRINT D$ "NOMONICO"
200 TEXT : HOME : HTAB 13 : PRINT "ULTIMAKER^ IV" : PRINT : PRINT : HTAB 12 : PRINT "BY^ DANNY^ POLLAK"
210 VTAB 20 : HTAB 5 : PRINT " INSERT^ COPY^ OF^ BRITANNIA^ DISK"
220 PRINT : HTAB 5 : PRINT "AND^ PRESS^ ANY^ KEY^ TO^ CONTINUE" ;
230 GET A$ : IF A$ = ESC$ THEN HOME : END
240 ONERR GOTO 2640
250 PRINT D$ "BLOAD^ ROST,A$800" D$ "BLOAD^ PRTY,A$A00"
260 POKE 216 ,Ø
270 HOME : HTAB 13 : PRINT "ULTIMAKER^ IV"
280 CA = 2048 : FOR X = Ø TO 7 : VTAB 5 + X : PRINT CHR$ (65 + X ) ")^ " ; :B = Ø
290 S = PEEK (CA + X * 32 + B ) : IF S THEN PRINT CHR$ (S ); :B = B + 1 : GOTO 290
300 PRINT : NEXT
310 PRINT : PRINT "POSSESSIONS:^ ^ ^ 
```

Table 2: Possession Bytes

0	Torches	(0-99)
1	Gems	(0-99)
2	Keys	(0-99)
3	Sextants	(0-99)
4	Stones	(0-255)
5	Runes	(0-255)
6	Items	(0-7)
7	3 Part Key	(0-7)
8-9	Food	(0-9999)
10	???	
11-12	Gold	(0-9999)
13	Horn	(0-1)
14	Wheel	(0-1)
15	Skull	(0-1)
16	???	
17	Cloth	(0-99)
18	Leather	(0-99)
19	Chain	(0-99)
20	Plate	(0-99)
21	Magic Chain	(0-99)
22	Magic Plate	(0-99)
23	Mystic Robe	(0-99)
24	???	
25	Staff	(0-99)
26	Dagger	(0-99)
27	Sling	(0-99)
28	Mace	(0-99)
29	Axe	(0-99)
30	Sword	(0-99)
31	Bow	(0-99)
32	Crossbow	(0-99)
33	Flaming Oil	(0-99)
34	Halberd	(0-99)
35	Magic Axe	(0-99)
36	Magic Sword	(0-99)
37	Magic Bow	(0-99)
38	Magic Wand	(0-99)
39	Mystic Sword	(0-99)
40-47	???	
48	Sulfer Ash	(0-99)
49	Genseng	(0-99)
50	Garlic	(0-99)
51	Spider Silk	(0-99)
52	Bloody Moss	(0-99)
53	Black Pearl	(0-99)
54	Nightshade	(0-99)
55	Mandrake	(0-99)
56	???	
57	Awaken	(0-99)
58	Blink	(0-99)
59	Cure	(0-99)
60	Dispel	(0-99)
61	Energy Field	(0-99)
62	Fireball	(0-99)
63	Gate Travel	(0-99)
64	Heal	(0-99)
65	Iceball	(0-99)
66	Jinx	(0-99)
67	Kill	(0-99)
68	Light	(0-99)
69	Magic Missile	(0-99)
70	Negate	(0-99)
71	Open	(0-99)
72	Protection	(0-99)
73	Quickness	(0-99)
74	Resurrect	(0-99)
75	Sleep	(0-99)
76	Tremor	(0-99)
77	Undead	(0-99)
78	View	(0-99)
79	Wind Change	(0-99)
80	Xit	(0-99)
81	Y (Up)	(0-99)
82	Z (Down)	(0-99)

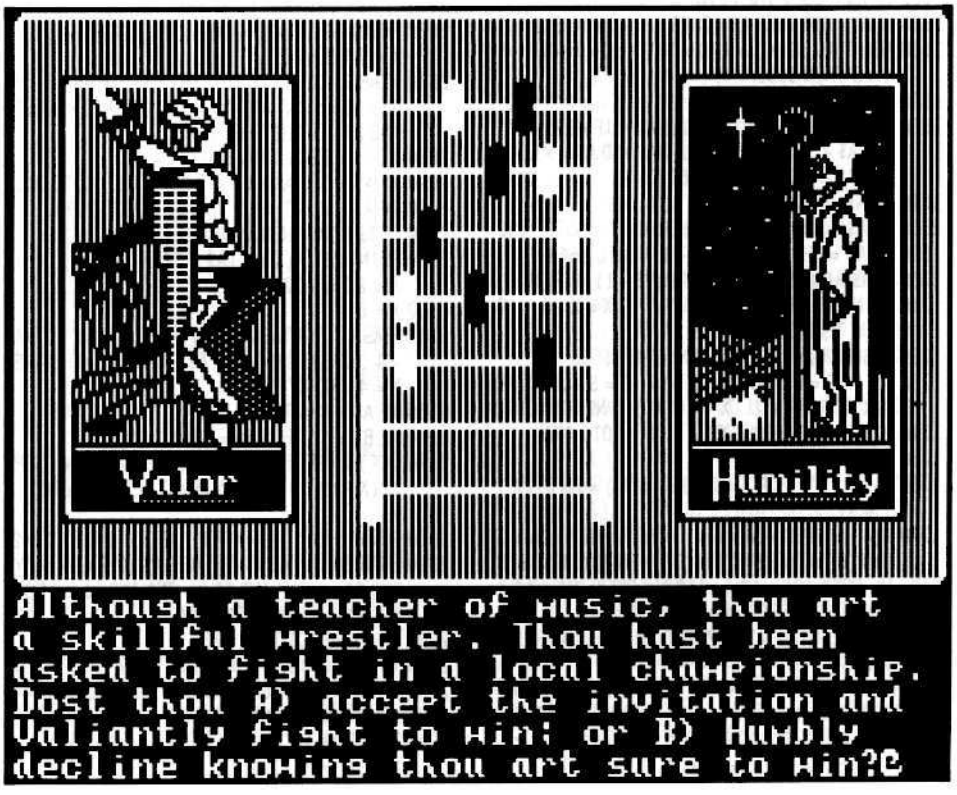
```
550 PRINT CH(X ) : NEXT : PRINT : PRINT
    "WEAPON========>" WE$(WE ) : PRINT
    "ARMOUR========>" AR$(AR )
560 REM EDIT NAME
570 VTAB 3 : HTAB 16 :A1$ = "^ " :A2$ = "Z" :MAX
    = 15 : GOSUB 2390 : IF A$ = CH$ THEN 570
580 IF B$ = "" THEN B$ = NAME$
590 VTAB 3 : HTAB 16 : PRINT B$ SPC( 16 - LEN (B$
    ) ) :NAME$ = B$
600 REM EDIT SEX
610 VTAB 4 : HTAB 16 : GET A$ : IF A$ < > ESC$ AND
    A$ < > "M" AND A$ < > "F" AND A$ < > CM$ AND
    A$ < > CH$ THEN 610
620 IF A$ = ESC$ THEN 270
630 IF A$ = CH$ THEN 570
640 IF A$ = CM$ THEN 670
650 SEX$ = "FEMALE" : IF A$ = "M" THEN SEX$ =
    "MALE^ ^ "
660 PRINT SEX$
670 REM EDIT TYPE
680 VTAB 5 : HTAB 16 :A1$ = "0" :A2$ = "7" :MAX =
    1 : GOSUB 2390 : IF A$ = CH$ THEN 610
690 IF B$ = "" THEN 720
700 TYPE = VAL (B$ ) : PRINT CH$Z$(TYPE ) SPC( 10 )
710 REM EDIT HEALTH
720 VTAB 6 : HTAB 16 : GET A$ : IF A$ = ESC$ THEN 270
730 IF A$ = CM$ THEN 820
740 IF A$ = CH$ THEN 680
750 IF A$ = "G" THEN HEALTH = 199 :X = 0 : GOTO 800
760 IF A$ = "D" THEN HEALTH = 196 :X = 1 : GOTO 800
770 IF A$ = "P" THEN HEALTH = 208 :X = 2 : GOTO 800
780 IF A$ = "S" THEN HEALTH = 211 :X = 3 : GOTO 800
790 GOTO 720
800 PRINT H$(X ) SPC( 10 )
810 REM EDIT CHARACTERISTICS
820 X = 0
830 VTAB 8 + X : HTAB 16 :MAX = 2 :A1$ = "0" :A2$
    = "9" : GOSUB 2390 : IF A$ = CH$ AND X = 0 THEN
    720
840 IF A$ = CH$ THEN X = X - 1 : GOTO 830
850 IF B$ = "" THEN B$ = STR$ (CH(X ) )
860 CH(X ) = VAL (B$ ) : HTAB 16 : PRINT CHR$ (48
    * (CH(X ) < 10 ) )CH(X ) :X = X + 1 : IF X <
    4 THEN 830
870 REM EDIT 4 DIGIT CHARS.
880 VTAB 9 + X : HTAB 16 :MAX = 4 :A1$ = "0" :A2$
    = "9" : GOSUB 2390 : IF A$ = CH$ AND X = 4 THEN
    X = 3 : GOTO 830
890 IF A$ = CH$ THEN X = X - 1 : GOTO 880
900 IF B$ = "" THEN B$ = STR$ (CH(X ) )
910 HTAB 16 :CH(X ) = VAL (B$ ) : IF CH(X ) < 1000
    THEN PRINT LEFT$ ( "000" ,4 - LEN ( STR$
    (CH(X ) ) ) );
920 PRINT CH(X ) :X = X + 1 : IF X < 7 THEN 880
930 REM EDIT WEAPON
940 VTAB 17 : HTAB 16 :MAX = 1 :A1$ = "A" :A2$ =
    "P" : GOSUB 2390 : IF A$ = CH$ THEN X = 6 :
    GOTO 880
950 IF B$ = "" THEN 980
960 WE = ASC (B$ ) - 65 : PRINT CH$WE$(WE ) SPC(
    10 )
970 REM EDIT ARMOUR
980 VTAB 18 : HTAB 16 :MAX = 1 :A1$ = "A" :A2$ =
    "H" : GOSUB 2390 : IF A$ = CH$ THEN 940
990 IF B$ = "" THEN 1010
1000 AR = ASC (B$ ) - 65 : PRINT CH$AR$(AR ) SPC(
    10 )
1010 GOSUB 2500 : IF A$ = "N" THEN 570
1020 REM UPDATE CHARACTER
1030 FOR X = 1 TO LEN (NAME$ ) : POKE CA + X - 1 ,
    ASC ( MID$ (NAME$ ,X ,1 ) ) : NEXT : POKE CA
    + X - 1 ,0
1040 POKE CA + 16 ,23 + 69 * ( LEFT$ (SEX$ ,1 )
    = "M" )
1050 POKE CA + 17 ,TYPE : POKE CA + 18 ,HEALTH
1060 FOR X = 0 TO 3 : POKE CA + 19 + X , INT (CH(X
    ) / 10 ) * 16 + (CH(X ) - INT (CH(X ) / 10
    ) * 10 ) : NEXT
1070 FOR X = 4 TO 6 : POKE CA + 24 + (X - 4 ) * 2
    , INT (( INT (CH(X ) / 1000 ) * 1600 + CH(X
    ) - INT (CH(X ) / 1000 ) * 1000 ) / 100 )
1080 POKE CA + 25 + (X - 4 ) * 2 , INT ((CH(X )
    - INT (CH(X ) / 100 ) * 100 ) / 10 ) * 16
    + INT (CH(X ) - INT (CH(X ) / 10 ) * 10 ) :
    NEXT
1090 POKE CA + 30 ,WE : POKE CA + 31 ,AR : GOTO 270
1100 REM WEAPONS & ARMOUR
1110 CA = 2329 : FOR X = 0 TO 6 :AR(X + 1 ) = FN B1(X
    ) : NEXT
1120 FOR X = 1 TO 15 :WE(X ) = FN B1(7 + X ) : NEXT
1130 HOME : HTAB 12 : PRINT "WEAPONS^ &^ ARMOUR"
    : PRINT
1140 FOR X = 1 TO 15 : PRINT WE$(X ) "^ " CHR$ (48
    * (WE(X ) < 10 ) )WE(X ) : NEXT
1150 VTAB 3 : FOR X = 1 TO 7 : HTAB 24 : PRINT AR$(X
    ) "^ " CHR$ (48 * (AR(X ) < 10 ) )AR(X ) :
    NEXT
1160 X = 1
1170 VTAB 2 + X : HTAB 14 :A1$ = "0" :A2$ = "9" :MAX
    = 2 : GOSUB 2390 : IF A$ = CH$ AND X = 1 THEN
    1170
1180 IF A$ = CH$ THEN X = X - 1 : GOTO 1170
1190 IF B$ = "" THEN B$ = STR$ (WE(X ) )
1200 HTAB 14 :WE(X ) = VAL (B$ ) : PRINT CHR$ (48
    * (WE(X ) < 10 ) )WE(X ) :X = X + 1 : IF X <
    16 THEN 1170
1210 X = 1
1220 VTAB 2 + X : HTAB 36 :A1$ = "0" :A2$ = "9" :MAX
    = 2 : GOSUB 2390 : IF A$ = CH$ AND X = 1 THEN
    X = 15 : GOTO 1170
1230 IF A$ = CH$ THEN X = X - 1 : GOTO 1220
1240 IF B$ = "" THEN B$ = STR$ (AR(X ) )
1250 HTAB 36 :AR(X ) = VAL (B$ ) : PRINT CHR$ (48
    * (AR(X ) < 10 ) )AR(X ) :X = X + 1 : IF X <
    8 THEN 1220
1260 GOSUB 2500 : IF A$ = "N" THEN 1160
1270 FOR X = 0 TO 6 : POKE CA + X , INT (AR(X + 1 )
    / 10 ) * 16 + (AR(X + 1 ) - INT (AR(X + 1
    ) / 10 ) * 10 ) : NEXT
1280 FOR X = 1 TO 15 : POKE CA + 7 + X , INT (WE(X
    ) / 10 ) * 16 + (WE(X ) - INT (WE(X ) / 10
    ) * 10 ) : NEXT : GOTO 270
1290 REM REAGENTS & MIXTURES
1300 CA = 2360 : FOR X = 0 TO 7 :RE(X ) = FN B1(X
    ) : NEXT : FOR X = 0 TO 25 :MI(X ) = FN B1(8
    + X ) : NEXT
1310 HOME : HTAB 10 : PRINT "REAGENTS^ &^
    MIXTURES" : PRINT
1320 FOR X = 0 TO 7 : PRINT RE$(X ); : HTAB 15 :
    PRINT CHR$ (48 * (RE(X ) < 10 ) )RE(X ) :
    NEXT
1330 VTAB 13 : FOR X = 0 TO 10 : PRINT MI$(X );; :
    HTAB 15 : PRINT CHR$ (48 * (MI(X ) < 10 )
    )MI(X ) : NEXT
1340 VTAB 3 : FOR X = 11 TO 25 : HTAB 23 : PRINT
    MI$(X );; : HTAB 37 : PRINT CHR$ (48 * (MI(X
    ) < 10 ) )MI(X ) : NEXT
1350 X = 0
1360 VTAB 3 + X : HTAB 15 :A1$ = "0" :A2$ = "9" :MAX
    = 2 : GOSUB 2390 : IF A$ = CH$ AND X = 0 THEN
    1360
1370 IF A$ = CH$ THEN X = X - 1 : GOTO 1360
1380 IF B$ = "" THEN B$ = STR$ (RE(X ) )
1390 RE(X ) = VAL (B$ ) : HTAB 15 : PRINT CHR$ (48
    * (RE(X ) < 10 ) )RE(X ) :X = X + 1 : IF X <
    8 THEN 1360
```

```
1400 X = 0
1410 VTAB 13 + X : HTAB 15 :A1$ = "0" :A2$ = "9"
     :MAX = 2 : GOSUB 2390 : IF A$ = CH$ AND X = 0
     THEN X = 7 : GOTO 1360
1420 IF A$ = CH$ THEN X = X - 1 : GOTO 1410
1430 IF B$ = "" THEN B$ = STR$ (MI(X))
1440 MI(X) = VAL (B$) : HTAB 15 : PRINT CHR$ (48
     * (MI(X) < 10))MI(X) :X = X + 1 : IF X <
     11 THEN 1410
1450 VTAB X - 8 : HTAB 37 :A1$ = "0" :A2$ = "9" :MAX
     = 2 : GOSUB 2390 : IF A$ = CH$ AND X = 11 THEN
     X = 10 : GOTO 1410
1460 IF A$ = CH$ THEN X = X - 1 : GOTO 1450
1470 IF B$ = "" THEN B$ = STR$ (MI(X))
1480 MI(X) = VAL (B$) : HTAB 37 : PRINT CHR$ (48
     * (MI(X) < 10))MI(X) :X = X + 1 : IF X <
     26 THEN 1450
1490 GOSUB 2500 : IF A$ = "N" THEN 1350
1500 FOR X = 0 TO 7 : POKE (CA + X), INT (RE(X)
     / 10) * 16 + (RE(X) - INT (RE(X) / 10
     ) * 10) : NEXT
1510 FOR X = 0 TO 25 : POKE (CA + 8 + X), INT (MI(X
     ) / 10) * 16 + (MI(X) - INT (MI(X) / 10
     ) * 10) : NEXT : GOTO 270
1520 REM STONES & RUNES
1530 CA = 2316 :S = PEEK (CA) :R = PEEK (CA + 1)
1540 FOR X = 0 TO 7 :R(X) = (R / 2 <> INT (R / 2
     )) :S(X) = (S / 2 <> INT (S / 2)) :R = INT
     (R / 2) :S = INT (S / 2) : NEXT
1550 HOME : HTAB 13 : PRINT "STONES^ &^ RUNES" :
     PRINT
1560 FOR X = 0 TO 7 : HTAB 5 : PRINT S$(X); : HTAB
     14 : PRINT CHR$ (78 + 11 * S(X)) : NEXT
1570 VTAB 3 : FOR X = 0 TO 7 : HTAB 22 : PRINT R$(X
     ); : HTAB 35 : PRINT CHR$ (78 + 11 * R(X
     )) : NEXT
1580 X = 0
1590 VTAB 3 + X : HTAB 14 : GET A$ : IF A$ <> "Y"
     AND A$ <> "N" AND A$ <> CM$ AND A$ <> ESC$
     AND A$ <> CH$ THEN 1590
1600 IF A$ = CH$ AND X = 0 THEN 1590
1610 IF A$ = CH$ THEN X = X - 1 : GOTO 1590
1620 IF A$ = ESC$ THEN 270
1630 IF A$ = CM$ THEN A$ = CHR$ (78 + 11 * S(X))
1640 PRINT A$ :S(X) = (A$ = "Y") :X = X + 1 : IF
     X < 8 THEN 1590
1650 X = 0
1660 VTAB 3 + X : HTAB 35 : GET A$ : IF A$ <> "Y"
     AND A$ <> "N" AND A$ <> CM$ AND A$ <> ESC$
     AND A$ <> CH$ THEN 1660
1670 IF A$ = ESC$ THEN 270
1680 IF A$ = CH$ AND X = 0 THEN X = 7 : GOTO 1590
1690 IF A$ = CH$ THEN X = X - 1 : GOTO 1660
1700 IF A$ = CM$ THEN A$ = CHR$ (78 + 11 * R(X))
1710 PRINT A$ :R(X) = (A$ = "Y") :X = X + 1 : IF
     X < 8 THEN 1660
1720 GOSUB 2500 : IF A$ = "N" THEN 1580
1730 S = 0 :R = 0 : FOR X = 0 TO 7 :S = S + 2 ^ X *
     S(X) :R = R + 2 ^ X * R(X) : NEXT
1740 POKE CA ,S : POKE CA + 1 ,R : GOTO 270
1750 REM OTHER ITEMS
1760 CA = 2312 : FOR X = 0 TO 3 : I(X) = FN B1(X)
     : NEXT
1770 I(4) = PEEK (CA + 6) : I(5) = PEEK (CA + 7
     ) : I(6) = FN B2(8) : I(7) = FN B2(11) : I(8
     ) = PEEK (CA + 13) : I(9) = PEEK (CA + 14)
     : I(10) = PEEK (CA + 15)
1780 HOME : HTAB 11 : PRINT "MISCELLANEOUS^
     ITEMS" : PRINT
1790 FOR X = 0 TO 3 : PRINT I$(X); : HTAB 10 : PRINT
     CHR$ (48 * (I(X) < 10))I(X) : NEXT
1800 S = I(4) : FOR X = 1 TO 3 :N(X) = (S / 2 <>
     INT (S / 2)) :S = INT (S / 2) : NEXT
1810 PRINT : PRINT "BELL" ; : HTAB 10 : PRINT CHR$
     (78 + 11 * N(1))
1820 PRINT "BOOK" ; : HTAB 10 : PRINT CHR$ (78 +
     11 * N(2))
1830 PRINT "CANDLE" ; : HTAB 10 : PRINT CHR$ (78
     + 11 * N(3))
1840 S = I(5) : FOR X = 4 TO 6 :N(X) = (S / 2 <>
     INT (S / 2)) :S = INT (S / 2) : NEXT
1850 PRINT : PRINT "3^ PT^ KEY:"
1860 PRINT "TRUTH" ; : HTAB 10 : PRINT CHR$ (78
     + 11 * N(4))
1870 PRINT "LOVE" ; : HTAB 10 : PRINT CHR$ (78 +
     11 * N(5))
1880 PRINT "COURAGE" ; : HTAB 10 : PRINT CHR$ (78
     + 11 * N(6))
1890 FOR X = 4 TO 5 : VTAB X - 1 : HTAB 20 : PRINT
     I$(X); : HTAB 28 :S = 4 - LEN (STR$ (I(X +
     2))) : IF S THEN PRINT LEFT$ ("000",S);
1900 PRINT I(X + 2) : NEXT
1910 PRINT : HTAB 20 : PRINT "HORN" ; : HTAB 28
     : PRINT CHR$ (78 + 11 * I(8))
1920 HTAB 20 : PRINT "WHEEL" ; : HTAB 28 : PRINT
     CHR$ (78 + 11 * I(9))
1930 HTAB 20 : PRINT "SKULL" ; : HTAB 28 : PRINT
     CHR$ (78 + 11 * I(10))
1940 N = PEEK (2575) : PRINT : HTAB 20 : PRINT "#^
     IN^ PARTY" ; : HTAB 31 : PRINT N
1950 X = 0
1960 VTAB 3 + X : HTAB 10 :A1$ = "0" :A2$ = "9" :MAX
     = 2 : GOSUB 2390 : IF A$ = CH$ AND X = 0 THEN
     1960
1970 IF A$ = CH$ THEN X = X - 1 : GOTO 1960
1980 IF B$ = "" THEN B$ = STR$ (I(X))
1990 I(X) = VAL (B$) : HTAB 10 : PRINT CHR$ (48
     * (I(X) < 10))I(X) :X = X + 1 : IF X < 4 THEN
     1960
2000 X = 1
2010 VTAB 7 + X : HTAB 10 : GET A$ : IF A$ <> "Y"
     AND A$ <> "N" AND A$ <> CM$ AND A$ <> ESC$
     AND A$ <> CH$ THEN 2010
2020 IF A$ = ESC$ THEN 270
2030 IF A$ = CH$ AND X = 1 THEN X = 3 : GOTO 1960
2040 IF A$ = CH$ THEN X = X - 1 : GOTO 2010
2050 IF A$ = CM$ THEN A$ = CHR$ (78 + 11 * N(X))
2060 PRINT A$ :N(X) = (A$ = "Y") :X = X + 1 : IF
     X < 4 THEN 2010
2070 VTAB 9 + X : HTAB 10 : GET A$ : IF A$ <> "Y"
     AND A$ <> "N" AND A$ <> CM$ AND A$ <> ESC$
     AND A$ <> CH$ THEN 2070
2080 IF A$ = ESC$ THEN 270
2090 IF A$ = CH$ AND X = 4 THEN X = 3 : GOTO 2010
2100 IF A$ = CH$ THEN X = X - 1 : GOTO 2070
2110 IF A$ = CM$ THEN A$ = CHR$ (78 + 11 * N(X))
2120 PRINT A$ :N(X) = (A$ = "Y") :X = X + 1 : IF
     X < 7 THEN 2070
2130 X = 6
2140 VTAB X - 3 : HTAB 28 :A1$ = "0" :A2$ = "9" :MAX
     = 4 : GOSUB 2390 : IF A$ = CH$ AND X = 6 THEN
     X = 4 : GOTO 2070
2150 IF A$ = CH$ THEN X = X - 1 : GOTO 2140
2160 IF B$ = "" THEN B$ = STR$ (I(X))
2170 I(X) = VAL (B$) : HTAB 28 :S = 4 - LEN (STR$
     (I(X))) : IF S THEN PRINT LEFT$ ("000",S
     );
2180 PRINT I(X) :X = X + 1 : IF X < 8 THEN 2140
2190 HTAB 28 : VTAB X - 2 : GET A$ : IF A$ <> "Y"
     AND A$ <> "N" AND A$ <> CM$ AND A$ <> ESC$
     AND A$ <> CH$ THEN 2190
2200 IF A$ = ESC$ THEN 270
2210 IF A$ = CH$ AND X = 8 THEN X = 7 : GOTO 2140
2220 IF A$ = CH$ THEN X = X - 1 : GOTO 2190
2230 IF A$ = CM$ THEN A$ = CHR$ (78 + 11 * I(X))
2240 PRINT A$ :I(X) = (A$ = "Y") :X = X + 1 : IF
     X < 11 THEN 2190
2250 VTAB 10 : HTAB 31 :A1$ = "1" :A2$ = "8" :MAX
     = 1 : GOSUB 2390 : IF A$ = CH$ THEN X = 10 :
     GOTO 2190
2260 IF B$ = "" THEN B$ = STR$ (N)
2270 N = VAL (B$) : GOSUB 2500 : IF A$ = "N" THEN
     1950
2280 FOR X = 0 TO 3 : POKE CA + X , INT (I(X) / 10
     ) * 16 + (I(X) - INT (I(X) / 10) * 10
     ) : NEXT
2290 POKE CA + 6 ,1 * N(1) + 2 * N(2) + 4 *
     N(3)
2300 POKE CA + 7 ,1 * N(4) + 2 * N(5) + 4 *
     N(6)
2310 POKE CA + 8 , INT (( INT (I(6) / 1000) *
     1600 + I(6) - INT (I(6) / 1000) * 1000 )
     / 100)
2320 POKE CA + 9 , INT ((I(6) - INT (I(6) / 100
     ) * 100) / 10) * 16 + INT (I(6) - INT
     (I(6) / 10) * 10)
2330 POKE CA + 11 , INT (( INT (I(7) / 1000) *
     1600 + I(7) - INT (I(7) / 1000) * 1000 )
     / 100)
2340 POKE CA + 12 , INT ((I(7) - INT (I(7) / 100
     ) *100) / 10) * 16 + INT (I(7) - INT
     (I(7) / 10) * 10)
2350 FOR X = 8 TO 10 : POKE CA + X + 5 , I(X) : NEXT
     : POKE 2575 ,N : GOTO 270
2360 REM SAVE CHARACTERS
2370 PRINT D$ "BSAVE^ ROST,A$800,L$200" D$
     "BSAVE^ PRTY,A$A00,L$20" : GOTO 270
2380 REM GET INPUT
2390 B$ = ""
2400 GET A$
2410 IF (A$ < A1$ OR A$ > A2$) AND (A$ <> ESC$ AND
     A$ <> CM$ AND A$ <> CH$ AND A$ <> CU$) THEN
     2400
2420 IF A$ = ESC$ THEN POP : GOTO 270
2430 IF A$ = CM$ OR (A$ = CH$ AND B$ = "") THEN
     RETURN
2440 IF A$ = CH$ AND LEN (B$) = 1 THEN B$ = "" :
     PRINT A$; : GOTO 2400
2450 IF A$ = CH$ THEN PRINT A$; :B$ = LEFT$ (B$
     , LEN (B$) - 1) : GOTO 2400
2460 IF A$ = CU$ THEN A$ = CHR$ ( PEEK ( PEEK (40
     ) + PEEK (41) * 256 + PEEK (36) ) - 128
     ) : GOTO 2410
2470 B$ = B$ + A$ : PRINT A$; : IF LEN (B$) = MAX
     THEN RETURN
2480 GOTO 2400
2490 REM OK PROMPT
2500 VTAB 20 : HTAB 30 : PRINT "OK^ ?" CH$; : GET
     A$ : IF A$ <> "Y" AND A$ <> "N" THEN 2500
2510 HTAB 30 : PRINT "^^^^^" ; : RETURN
2520 DATA MAGE ,BARD ,FIGHTER ,DRUID ,TINKER
     ,PALADIN ,RANGER ,SHEPERD
2530 DATA GOOD ,DEAD ,POISONED ,SLEEPING
2540 DATA HANDS ,STAFF ,DAGGER ,SLING ,MACE
     ,AXE ,SWORD ,BOW ,CROSSBOW ,FLAMING^ OIL
     ,HALBERD ,MAGIC^ AXE ,MAGIC^ SWORD
     ,MAGIC^ BOW ,MAGIC^ WAND ,MYSTIC^ SWORD
2550 DATA NONE ,CLOTH ,LEATHER ,CHAIN ,PLATE
     ,MAGIC^ CHAIN ,MAGIC^ PLATE ,MYSTIC^ ROBE
2560 DATA SULFER^ ASH ,GINSENG ,GARLIC ,SPIDER^
     SILK ,BLOOD^ MOSS ,BLACK^ PEARL
     ,NIGHTSHADE ,MANDRAKE^ ROOT
2570 DATA AWAKEN ,BLINK ,CURE ,DISPEL
     ,ENERGYFIELD ,FIREBALL ,GATE^ TRAVEL
     ,HEAL ,ICEBALL ,JINX ,KILL ,LIGHT ,MAGIC^
     MISSILE ,NEGATE ,OPEN ,PROTECTION
     ,QUICKNESS ,RESURRECT ,SLEEP ,TREMOR
     ,UNDEAD ,VIEW ,WIND^ CHANGE ,XIT ,Y^ (UP)
     ,Z^ (DOWN)
2580 DATA BLACK ,WHITE ,PURPLE ,ORANGE ,GREEN
     ,RED ,YELLOW ,BLUE
```

```
2590 DATA HUMILITY ,SPIRITUALITY ,HONOR
     ,SACRAFICE ,JUSTICE ,VALOR ,COMPASSION
     ,HONESTY
2600 DATA TORCHES ,GEMS ,KEYS ,SEXTANTS ,FOOD
     ,GOLD
2610 DATA STRENGTH======> ,DEXTERITY=====>
     ,INTELLEGENCE==> ,MAGIC^ POINTS==>
2620 DATA HIT^ POINTS====> ,HIT^ MAXIMUM===>
     ,EXPERIENCE====>
2630 REM ERROR
2640 IF PEEK (222) = 6 THEN HOME : VTAB 10 : HTAB
     6 : PRINT CG$ "THIS^ IS^ NOT^ A^ BRITANNIA^
     DISK" : FOR X = 1 TO 3000 : NEXT X : POKE -
     16368 ,0 : GOTO 200
2650 PRINT CG$CG$CG$ "ERR" : STOP
```

Ultimaker checksums

```
10   - $BADD      1340 - $BAFA
20   - $9B13      1350 - $9AFC
30   - $4D3B      1360 - $F20A
40   - $AD92      1370 - $A3B4
50   - $C899      1380 - $66FB
60   - $FF65      1390 - $4A85
70   - $A3BF      1400 - $D7A7
80   - $A900      1410 - $426B
90   - $0DB6      1420 - $90B1
100  - $6662      1430 - $5D2F
110  - $DE94      1440 - $73FC
120  - $4DB8      1450 - $85BA
130  - $1791      1460 - $C93A
140  - $0E40      1470 - $ACE0
150  - $ECFE      1480 - $9BDE
160  - $CA9E      1490 - $0D34
170  - $C85B      1500 - $534F
180  - $4958      1510 - $EC72
190  - $F38C      1520 - $A0FC
200  - $8314      1530 - $1E4E
210  - $2678      1540 - $4222
220  - $AF01      1550 - $BA00
230  - $FB75      1560 - $1C5B
240  - $4BDE      1570 - $D9B7
250  - $D520      1580 - $42F7
260  - $4CC4      1590 - $A889
270  - $A6DD      1600 - $0858
280  - $3C7C      1610 - $2133
290  - $7B84      1620 - $F108
300  - $A5FE      1630 - $B5B7
310  - $46B6      1640 - $4365
320  - $6161      1650 - $3256
330  - $8F93      1660 - $1998
340  - $6A62      1670 - $F6AD
350  - $79C7      1680 - $5603
360  - $5D16      1690 - $1E6D
370  - $BA58      1700 - $30AE
380  - $0E2F      1710 - $1F29
390  - $4BFF      1720 - $D992
400  - $B07D      1730 - $E62B
410  - $16B0      1740 - $C07D
420  - $124B      1750 - $6C22
430  - $5380      1760 - $75F6
440  - $8C6A      1770 - $860B
450  - $1A55      1780 - $4E02
460  - $05A6      1790 - $4BAF
470  - $D4F5      1800 - $3CC7
480  - $C520      1810 - $E4C3
490  - $F59E      1820 - $F5EA
500  - $03F0      1830 - $E5F8
510  - $228A      1840 - $0625
520  - $FA1F      1850 - $D86A
530  - $5F18      1860 - $5DC1
540  - $7A85      1870 - $CA90
550  - $96AB      1880 - $DCA1
560  - $C03B      1890 - $8FC1
570  - $DF72      1900 - $F909
580  - $2A33      1910 - $1979
590  - $2C25      1920 - $6652
600  - $0691      1930 - $2228
610  - $BC9F      1940 - $2B05
620  - $2C1C      1950 - $26D5
630  - $67E2      1960 - $728A
640  - $8CB4      1970 - $76C9
650  - $C395      1980 - $98C9
660  - $F627      1990 - $7950
670  - $C4C8      2000 - $3C8C
680  - $FC07      2010 - $3B35
690  - $047A      2020 - $BA9A
700  - $8335      2030 - $4F8A
710  - $56D1      2040 - $732D
720  - $2E38      2050 - $A433
730  - $C68E      2060 - $3733
740  - $EF55      2070 - $D5D1
750  - $B35C      2080 - $F5E5
760  - $2882      2090 - $25C6
770  - $40FF      2100 - $3B8A
780  - $251D      2110 - $6448
790  - $2ACA      2120 - $CC9E
800  - $2408      2130 - $26A6
810  - $153F      2140 - $079E
820  - $C862      2150 - $F7DC
830  - $23E4      2160 - $F897
840  - $DF47      2170 - $BF13
850  - $1E80      2180 - $D3E7
860  - $3E56      2190 - $87C2
870  - $D34F      2200 - $DDA6
880  - $8650      2210 - $AB57
890  - $E0C8      2220 - $24E4
900  - $2293      2230 - $8CA7
910  - $9A43      2240 - $661F
920  - $867C      2250 - $4899
930  - $81A9      2260 - $E1D9
940  - $2108      2270 - $BC05
950  - $E372      2280 - $E665
960  - $2562      2290 - $D32D
970  - $5F5E      2300 - $7637
980  - $9DE2      2310 - $F95C
990  - $A036      2320 - $106E
1000 - $0651      2330 - $38CB
1010 - $E7C8      2340 - $6C9D
1020 - $95B7      2350 - $7690
1030 - $642E      2360 - $2432
1040 - $3CD1      2370 - $AE0D
1050 - $5439      2380 - $DD62
1060 - $7BB7      2390 - $8B51
1070 - $CA4C      2400 - $31BA
1080 - $433E      2410 - $729E
1090 - $84E9      2420 - $C289
1100 - $E525      2430 - $76FA
1110 - $B6A8      2440 - $449F
1120 - $F6D8      2450 - $D0D5
1130 - $42D6      2460 - $DA67
1140 - $73D1      2470 - $71B7
1150 - $D287      2480 - $8B21
1160 - $34D2      2490 - $B11C
1170 - $3C6D      2500 - $B3C5
1180 - $D044      2510 - $12E3
1190 - $1908      2520 - $57CE
1200 - $242B      2530 - $D18B
1210 - $D611      2540 - $5848
1220 - $9B83      2550 - $3E9F
1230 - $BF60      2560 - $1325
1240 - $31D5      2570 - $749C
1250 - $06AF      2580 - $A82D
1260 - $2595      2590 - $8540
1270 - $7C2C      2600 - $A634
1280 - $73F9      2610 - $337E
1290 - $1780      2620 - $FABF
1300 - $B116      2630 - $2810
1310 - $F152      2640 - $0CB7
1320 - $CB53      2650 - $C511
1330 - $3614
```



The nearly complete softkey for...

SSI's RDOS

by Mike McConnell

Strategic Simulations, Inc.
883 Steirlin Rd., Bldg. A-200
Mountain View, CA 94093

Requirements:
48K Apple][(and up)
A Strategic Simulations RDOS game
Super IOB
A sector editor
A disk search program
Lots of blank disks
DOS 3.3 System Master
A 16K slave disk (optional)
DOS 3.2's RWTS

Editor's note: *It took a while, but finally, most of the information needed to convert Strategic Simulations' 13-sector RDOS format to a COPYAable and modifiable (under RDOS) form has been collected under one roof into a step-by-step method almost anyone can use. COMPUTIST invites those readers with additional information about RDOS and its protection(s) to submit their findings thus providing a welcome service to fellow readers. COMPUTIST would like to thank Clay Harrell for his role in this article.*

The Conversion

The first step in cracking these disks is getting them into DOS 3.3 format. To do this we use a highly modified Super IOB controller, the DOS 3.2 RWTS (RWTS.13), and RDOS.READ. You need to create RDOS.READ using RWTS.13 as shown below:

```
BLOAD RWTS.13,A$1900
CALL -151
1A76:D4
1A8B:B7
BSAVE RDOS.READ,A$1900,L$800
```

The format of most RDOS disks is normal DOS 3.2 with the address prologue bytes changed to D4 AA B7. A few RDOS disks use normal DOS 3.2 with no changes. The SSI controller can handle both types since you select which RWTS the program will use. To determine which format a particular RDOS disk uses you need a 16K slave disk or a way to reset into the monitor. Boot the protected RDOS disk and wait for the drive to stop. Insert your 16K slave disk and do the following, or simply enter the monitor if possible.

Hit Reset (the disk will boot and leave you in BASIC)

```
CALL -151
BC76
BC8B
```

The two numbers returned should be either D4 and B7, or D5 and B5 (normal) and will determine which RWTS you tell the controller to use.

Now that we know which RWTS to use we can make a copy of the RDOS disk in normal DOS 3.3 format using SUPER IOB. Type in the controller at the end of this article and save it. Install the controller by your favorite method and start Super IOB. RWTS.13 and RDOS READ should be on your Super IOB disk before starting. Be sure to write-protect the original disk before copying!!

Now you will have to type either "1" for D5 AA B5 format or "2" for D4 AA B7 format depending on which format the disk uses. We will assume that your disk is D4 AA B7 format so you need to type "2" and hit Return. The controller will load in the correct RWTS and continue normally (for a while anyway). Answer the disk prompts as you normally do, being sure to initialize the target disk using volume number 254. This makes sure that the three unused sectors in each track are zeroed so investigating the cracked disk is not harder than it has to be.

Now we come to the next major challenge of RDOS; it uses different sector interleaving from normal DOS 3.3. This means that the sectors read by the RWTS routine are not in the correct places. I have compensated for this as you will see when the sectors are written.

The following chart shows the sectors read by the RWTS and the actual locations they must be written to for RDOS to use them.

```
READ  - 0 1 2 3 4 5 6 7 8 9 A B C
WRITE - 0 7 E 6 D 5 C 4 B 3 A 2 9
```

As you can see, RDOS uses only 13 sectors per track so our normalized copy will have 16 sectors with 13 of them containing data. Sectors 1, 8, and F of each track are not used and should contain 00's. The WRITE table above is the order in which the sectors are written to our copy. Don't be alarmed when you see the sector numbers jumping around as they are written, this is normal. Now you should insert your disks and copy away!!

Creating RDOS 3.3

If you have a previously cracked copy of any SSI RDOS game, then all you need to do is copy track 0 from the cracked game to track 0 of the copy we made. Once you copy this track you still may have to remove the secondary protection and fix SSI.INIT to have a working copy.

Note: Versions of RDOS vary slightly and you may have to create RDOS 3.3 as shown below even if you have a previously cracked disk.

Now that we have the RDOS disk into normal DOS 3.3 format we must capture RDOS from the original and modify it to work with DOS 3.3-type disks. To do this we boot-trace the original disk, move the RDOS code out of the way, and boot a 48K Slave Disk to save it out.

```
CALL -151
9600<C600.C6FFM
96FA:98
9801:4C 59 FF
9600G
C0E8
```

Move Boot0 up and disassemble it if you wish. This is the famous "Brody-Loady" used by many publishers because they believe it is

a complicated boot (it's not). We simply modify it to jump to $9301, not $301 where the next step occurs.

```
9800<800.8FFM
9805:98
9843:93
9301:4C 59 FF
9600G
C0E8
```

Now we move the next step up to $9300 and modify it to go to the small ML program below. This routine checks the progress of Boot0 and enters the monitor when RDOS has been read in.

```
9300<300.3FFM
9343:4C 00 90
9000:A5 3E C9 5D D0 03 4C 5D
9008:02 4C 59 FF
9600G
C0E8
```

Now we move RDOS and then save it as follows:

```
7200<B200.BFFFM
```

Insert a 48K slave disk

```
C600G
BSAVE RDOS,A$7200,L$E00
```

Now we modify RDOS to work with our normalized RDOS disk. The procedure is fairly complicated so I will not go into great detail about it. What we are doing is substituting normal DOS 3.3 postnibble and prenibble routines along with some needed data. We also change the address marks to normal and do a few other things such as zeroing the primary and secondary data buffers.

If you wish you can make a text file out of the following, without the BLOAD and BSAVE commands but you need to add the following to make an EXEC of the file work.

Add **POKE 72,0** just before the CALL -151. Type in everything between the BLOAD and BSAVE commands. Add **48:D4** and **D43CG** to the end of the file and save it. Now you simply EXEC the file after loading RDOS and then save out the converted RDOS 3.3 file. Using an EXEC file greatly speeds the conversion of RDOS to RDOS 3.3.

```
BLOAD RDOS
CALL -151
7B00<B800.B829M
7B2A:00 N 7B2B<7B2A.7B68M

7B0A:BF N 7B0E:BF
7B11:BE
7B20:BF N 7B25:BF
7B89:04 48 68
7BA3:56
7BAF:BF N 7BC4:BF N 7BD7:BF

7CC1<B8C2.B8DBM
7C16:56
7C75:C9 D5
7C8B:96
7CCA:BE
7CD6:C0 00
7CDC:A5 2D 85 F9 A5 2E 85 FA
7CE4:A5 3E 85 FB A5 3F 85 FC
7CEC:00

7D96<BA96.BAFFM
7E00:00 N 7E01<7E00.7EFEM
7F00<7E00.7EFFM
7F70<BA29.BA68M
BSAVE RDOS 3.3,A$7200,L$E00
```

Creating the Boot Sector

Now we need to create the boot sector. You need a DOS 3.3 System Master and a little boot-tracing to get the needed sector.

```
BLOAD RDOS 3.3
```

Insert System Master disk

```
CALL -151
9600<C600.C6FFM
96FA:98
9801:4C 59 FF
9600G
C0E8
```

Now you need to move the boot sector up to $7200 and modify it to function with our copy. We set the high sector to $0D and the start of the program to $B300 (B2 + 1).

```
7200<800.8FFM
72FE:B2 0D
9D84G
```

Insert 48K Slave Disk

```
BSAVE RDOS 3.3,A$7200,L$E00
```

Now you should have a complete DOS 3.3 compatible version of RDOS. Next you need to write the sectors to the copy using the small ML program below:

Boot a 48K Slave Disk

```
BLOAD RDOS 3.3
CALL -151
300:A9 00 8D EB B7 8D EC B7
308:8D F0 B7 A9 0D 8D ED B7
310:A9 7F 8D F1 B7 A9 02 8D
318:F4 B7 A9 B7 A0 E8 20 B5
320:B7 CE F1 B7 CE ED B7 AD
328:ED B7 C9 FF D0 01 60 8D
330:ED B7 4C 1A 03
```

Check your typing against the following dis-assembly:

```
0300-  A9 00     LDA  #$00
0302-  8D EB B7  STA  $B7EB
0305-  8D EC B7  STA  $B7EC
0308-  8D F0 B7  STA  $B7F0
030B-  A9 0D     LDA  #$0D
030D-  8D ED B7  STA  $B7ED
0310-  A9 7F     LDA  #$7F
0312-  8D F1 B7  STA  $B7F1
0315-  A9 02     LDA  #$02
0317-  8D F4 B7  STA  $B7F4
031A-  A9 B7     LDA  #$B7
031C-  A0 E8     LDY  #$E8
031E-  20 B5 B7  JSR  $B7B5
0321-  CE F1 B7  DEC  $B7F1
0324-  CE ED B7  DEC  $B7ED
0327-  AD ED B7  LDA  $B7ED
032A-  C9 FF     CMP  #$FF
032C-  D0 01     BNE  $032F
032E-  60        RTS
032F-  8D ED B7  STA  $B7ED
0332-  4C 1A 03  JMP  $031A
```

```
BSAVE RDOS WRITER,A$300,L$35
```

Insert the copy

```
300G
```

The sectors will be written as follows on track 0 of the copy.

Sector -	Mem. Page
00 -	$B200
01 -	$B300
02 -	$B400
03 -	$B500
04 -	$B600
05 -	$B700
06 -	$B800
07 -	$B900
08 -	$BA00
09 -	$BB00
0A -	$BC00
0B -	$BD00
0C -	$BE00
0D -	$BF00

Removing the Secondary Protection

The disk will now boot but will hang partway or reboot because of the secondary protection we still need to remove. An exception to this is Computer Quarterback 1.1 which has no secondary protection!

We need to remove the secondary protection by using something like the Core Disk Searcher (Hardcore COMPUTIST No. 12) or any other disk search utility. The protection usually goes under the name of "QWERTY" or "@WERTY" and exists in three forms that I know of. Only one form of QWERTY/@WERTY should exist per disk.

The QWERTY routine is the most common and looks for the byte $EE to be at a certain location. To disable the QWERTY routine you need to search each disk for the bytes $49 EE D0.

Once you know where it is you need to disassemble the sector and make sure the routine looks like the one below.

```
31E-  BD 8E C0   LDA  $C08E,X
321-  BD 8C C0   LDA  $C08C,X
324-  10 FB      BPL  $0321
326-  49 EE      EOR  #$EE
328-  D0 E5      BNE  $030F
32A-  85 00      STA  $00
32C-  60
```

Change bytes $28-$29 of this sector to $A9 00 to circumvent this check.

A rare @WERTY file looks for an $AA following the address field and reboots if it is not found. Search for the sequence $49 AA D0. Change the found $D0 (and the next byte) to $A9 00

A newer technique uses spiral tracking to read in four half-tracks from track $20.5 to $22 and store the data at $1000-$1FFF. Most copy

programs will not copy this correctly because track bleeding will trash adjacent half-tracks as they are written. Data is Exclusive-ORed with its own address and the program reboots if an error is found. To disable this, just put an RTS ($60) at the beginning of @WERTY (the entry point is at $A0F0) and the whole routine is ignored.

Only one of these routines will be on each disk. On multi-disk games the same routine should occur on each disk but the locations may vary.

Fixing the INIT

After removing the secondary protection all that is left to do is modify the program that initializes data disks. The program is usually called SSI.INIT. The address marks need to be changed to the normal DOS 3.3 marks. You should search each disk for the bytes: 20 CC 09 A9 AA 20 CD 09

Compare the sector to the following disassembly to make sure it is correct.

```
8F4-  A9 D4      LDA  #$D4 (or D5)
8F6-  20 CC 09   JSR  $09CC
8F9-  A9 AA      LDA  #$AA
8FB-  20 CD 09   JSR  $09CD
8FE-  A9 B7      LDA  #$B7 (or B5)
```

You need to change byte $F5 to D5 and byte $FF to 96. Then write the sector back out to the disk. This is the last hurdle in cracking RDOS games. For some strange reason Questron (side 0) contains two SSI.INIT sectors even though the game never INIT's a disk!!

More Notes

Now you have a completely cracked (COPYAable) copy of your RDOS disk with one small problem. RDOS is very fast because it is DOS 3.2, but RDOS 3.3 is very slow because it is DOS 3.3 reading RDOS sector interleaving. This can be fixed by using Bag Of Tricks' INIT program and changing the sector skewing to "ascending 01" on all tracks. Remember to preserve the data if you already have the broken game on the disk.

If the original of the disk you just cracked is normally write-protected you should write-protect the copy we just made. As far as I know RDOS games never check for a write-protect label but this will prevent accidents.

You do not need to re-create RDOS 3.3 for each disk. You need only copy track 0 from any cracked RDOS game to a copy made with SUPER IOB. Then remove the secondary protection and fix SSI.INIT to fully crack the disk.

But: Some versions of RDOS differ so you may have to re-create RDOS 3.3 for some particular disks.

Removing the Reset Routine

The reset routine used by SSI is in RDOS and can be modified to enter the monitor on reset instead of rebooting the disk. The routine is located in various places (2 that I have seen) in page $B9 ($B900-B9FF). To remove the routine boot a sector editor and read in track 0, sector 7. Now you need to locate the bytes 20 2F FB 20 58 FC, which are the beginning of the reset routine. I found these bytes at $B9B0 or $B9D0.

Once you find these bytes you need to change the first three to 4C 59 FF and re-write the sector. This changes the reset routine to enter the monitor instead of rebooting the disk. Once in the monitor you need to change $D6, which is the RUN flag, to 00 before entering BASIC. When $D6 is more than 127 all commands RUN the current BASIC program. Before entering BASIC you must reset the flag with

D6:00

then use CTRL-C to enter BASIC.

Some RDOS disks modify the reset routine (Questron and others) just after RDOS has been read in. The above technique will still work but instead of waiting for the drive to stop you should hit reset just after RDOS has been loaded in. RDOS is loaded in about 4 seconds after the boot starts.

Sometimes you can remove the reset routine change by modifying the HELLO program. You need to remove all of the following POKES from the HELLO program:

```
POKE 1010,xxx
POKE 1011,xxx
POKE 1012,xxx
```

These POKES set the reset vector to enter the new reset routine, not our modifed old reset routine. The routine is usually loaded in directly before or after these POKES. Remember that these POKES can occur in more than one line. I found them in two lines in the Questron HELLO program (118 and 51).

Modifying RDOS Disks

Using a 16K Slave Disk and a small text file you can use RDOS 3.3 to modify the programs on each disk. The following is the text file you need to create using a normal word processor. Call it RDOS.FIX.

```
POKE72,0:CALL-151
3D0:4C B0 B9 4C 00 E0 4C FD
3D8:AA 4C 00 BA AD 0F 9D AC
3E0:0E 9D 60 AD C2 AA AC C1
3E8:AA 60 60 60 60 EA EA 4C
3F0:59 FA 59 FF 5A 4C 03 B3
3F8:4C 65 FF 4C 65 FF 65 00
48:D4
D43CG
```

To access any RDOS disk simply do the following:

Boot a 16K Slave Disk

```
BLOAD RDOS 3.3,A$B200
EXEC RDOS.FIX
```

Insert your cracked RDOS disk

The following commands are active in RDOS. All RDOS commands are accessed through the ampersand hook in page 3, which is set by the text file. All commands except &CAT require the file name to be enclosed in quotes.

RDOS COMMAND	DOS 3.3 COMMAND
&DEL "FILE"	DELETE FILE
&CAT	CATALOG DISK
&RECALL "FILE"	BLOAD FILE
&STORE "FILE"	BSAVE FILE
&LOAD "FILE"	LOAD FILE
&SAVE "FILE"	SAVE FILE
&RUN "FILE"	RUN FILE
&GOTO "FILE"	CHAIN FILE
&DEF "FILE"	OPEN FILE
&READ "FILE"	READ FILE
&PRINT "FILE"	WRITE FILE
&END "FILE"	CLOSE FILE

Using this technique you can modify any RDOS game, adding cheat routines or anything else you might need. Before re-saving any file you must first DELETE it using &DEL. The binary commands work differently from normal DOS 3.3 commands. You cannot use hex numbers when working with the binary commands (&RECALL, and &STORE). You must convert the length and address to decimal. Below are examples of commands that will work:

& RECALL "QWERTY" - BLOAD's where it was saved

& RECALL "QWERTY",32768 - BLOAD's at 32768 ($8000)

& STORE"QWERTY",32768,50 - save file at 32768 with 50 length

controller

```
1000 REM SSI CONTROLLER
1005 DIM ST$(13) : FOR X = 0 TO 12 : READ ST$(X
     ) : NEXT
1010 TK = 1 :ST = 0 :LT = 35 :CD = WR :DOS = 13
1020 T1 = TK : GOSUB 490 : GOSUB 360 : ONERR GOTO
     550
1030 GOSUB 430 : GOSUB 100 :ST = ST + 1 : IF ST <
     DOS THEN 1030
1040 IF BF THEN 1060
1050 ST = 0 :TK = TK + 1 : IF TK < LT THEN 1030
1060 GOSUB 490 :TK = T1 :S = 0 : GOSUB 360
1070 ST = VAL (ST$(S) ) : GOSUB 430 : GOSUB 100
     :S = S + 1 : IF S < DOS THEN 1070
1080 S = 0 :TK = TK + 1 : IF BF = 0 AND TK < LT THEN
     1070
1090 IF TK < LT THEN ST = 0 : GOTO 1020
1100 HOME : PRINT "EVERYTHING^ O.K.^ NO^ DOS^
     ON^ COPY" : END
5000 DATA 0,7,14,6,13,5,12,4,11,3,10,2,9
10010 HOME : VTAB 2 : PRINT "WHICH^ (1-2)^ " :
     PRINT : PRINT "1.^ RDOS^ (D5^ AA^ B5)" :
     PRINT : PRINT "2.^ RDOS^ (D4^ AA^ B7)"
10012 INPUT ST : IF ST < 1 OR ST > 2 THEN 10010
10014 IF ST = 1 THEN PRINT CHR$ (4 ) "BLOAD^
     RWTS.13,A$1900" : GOTO 10020
10016 PRINT CHR$ (4 ) "BLOAD^ RDOS.READ,A$1900"
```

controller checksums

```
1000 - $356B      1080 - $B44A
1005 - $BB0E      1090 - $DEB4
1010 - $3694      1100 - $0A0F
1020 - $4041      5000 - $A7A1
1030 - $5640      10010 - $3782
1040 - $9441      10012 - $E87D
1050 - $965E      10014 - $71C4
1060 - $83FB      10016 - $F332
1070 - $DB46
```

softkey for...

by Mike Saul

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

Requirements:
Super IOB
Fantavision
2 blank disks (or one double-sided)
A sector editor

Fantavision is a newly released program from Broderbund for making movies on the Apple. It offers smooth animation for up to eight different objects simultaneously. Utilizing a process known as "tweening", Fantavision does not even require the user to draw every frame. It will fill in up to sixty-four intermediate frames for each one the user creates. It is very user friendly, too, with its Macintosh-like pulldown menus and icons.

In fact, the package is outstanding in every respect but one: it is protected.

The Protection

Fantavision does allow the purchaser to make one backup copy, but in many cases one copy is not enough. Attempts to nibble copy the program were rather dismal. The copies would boot, but after two or three tracks had loaded, the system would just hang. To me this indicated that there was probably a nibble count hidden on the disk somewhere.

A preliminary examination of Fantavision with a nibble editor revealed two interesting things. First, the address epilog bytes had been changed from their usual DE AA to DE AB, and, second, track $22 did not have a normal sector structure as did the rest of the disk. Modified DOS marks are a common form of disk protection, and they usually do not present much of a problem. Track $22, however, was another story.

To get a better idea of what was actually happening during the boot process, I interrupted the boot with my copy card and started disassembling memory. Eventually I discovered that the offending nibble count was residing at $BEAD. A search for this code on the Fantavision disk, though, yielded nothing. It seems the folks at Broderbund decided to be tricky. The code for the nibble count actually resides on track $22. Just to make life harder for us, the code needed for the rest of the boot is there too. What actually happens is this: track $22 is read and decoded. Then the nibble count is executed. If the count is correct, the boot continues normally. Otherwise, the boot stops.

The Plan

Obviously a track which does not contain sectors cannot be COPYA'd so track $22 needed to be formatted normally. The code from that track had to somehow be placed into memory so that the boot could continue. The nibble count had to be disabled, and finally, the address marks needed to be normalized. These criteria form the basis for the softkey below.

The Softkey

1) Boot the computer with any normal DOS 3.3 disk and enter the monitor.

CALL -151

Move the boot ROM on the disk controller card, known as BOOT0, into RAM so that it can be modified.

9600<C600.C6FFM

Patch the moved code so that it will fall back into the monitor when BOOT1 (Track 0, sector 0) has been loaded.

96F9:59 FF

2) Put the Fantavision disk in drive one and execute the modified BOOT0.

9600G

Turn off the disk drive,

C0E8

and move BOOT1 to a safe place.

9700<800.8FFM

Change the moved BOOT0 so that it will jump to the moved BOOT1.

96F9:01 97

Make a change so BOOT1 will run at its new location,

9751:97

and patch a jump to the monitor into BOOT1.

977F:59 FF

3) Execute the boot again so that more of the program will be loaded into memory,

9600G

and turn off the drive.

C0E8

4) At this point everything we will need is in memory. Fantavision's decoded track $22 is sitting between $BC00 and $BFFF. We will only be using the code from $BE00 to $BFFF, however. The nibble count is at $BEAD. Two other routines worth examining are those at $B126 and $B011. If you disassemble memory at these locations, you might find they look somewhat like parts of a normal DOS. This is in fact what they are. Respectively, the routines locate a sector's address fields and read a sector of data.

These are the routines we will be using later to read from our normalized disk. The code which was responsible for reading track $22 is between $B500 and $B568 if you want to look at that while you are here. Now, move the code we need down to a safe place in memory,

1900<BE00.BFFFM

and disable the nibble count.

19AB:0F BF

5) Put your Super IOB disk (make sure it's a slave) in drive 1 and reboot.

6P

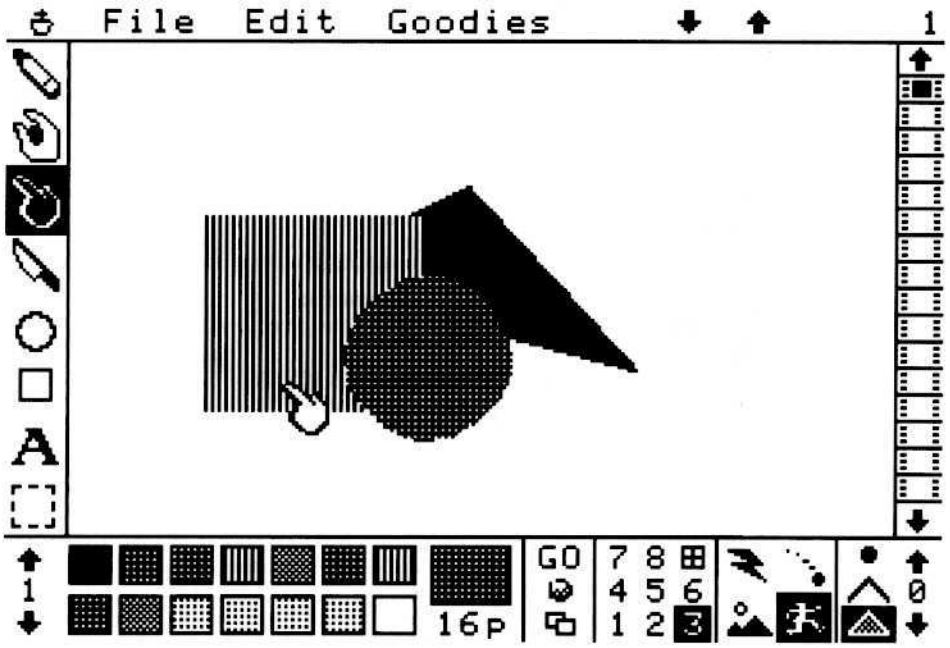
Now, it's time to capture the code we need.

BSAVE FVSTUFF,A$1900,L$1FF

6) Install the Fantavision controller given here into Super IOB and RUN the resulting program.

7) When Super IOB has done its work, boot up your favorite sector editor and enter the hexdump below into track $15, sector $05 of the copy. Start at byte $00 and just type right over the existing bytes. The code you are entering will supercede Fantavision's old track $22 read routine and allow normal sectors to be read.

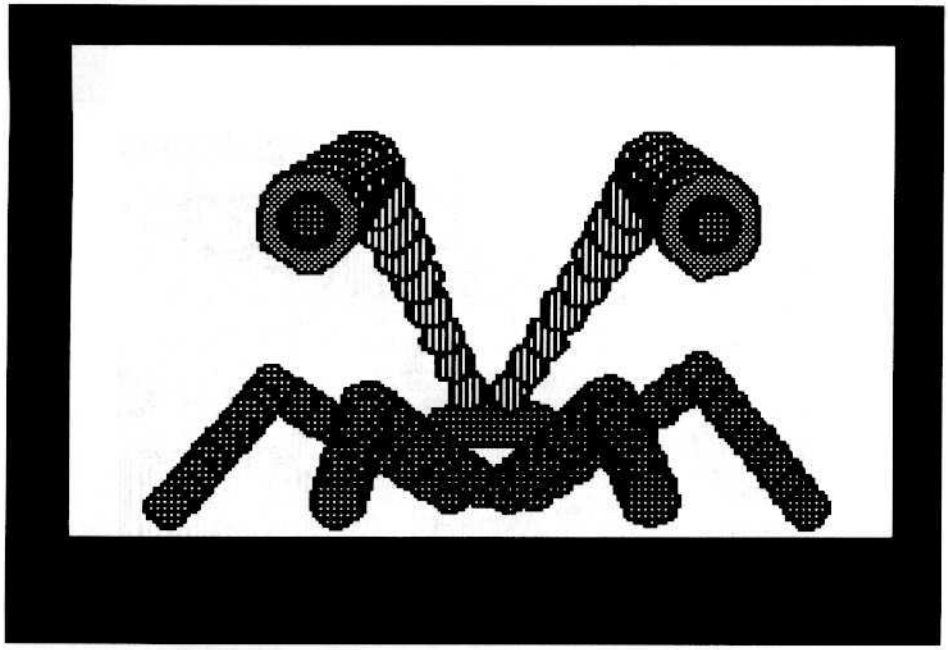
```
$xx00:  A9 22 20 09 B0 A2 60 20
$xx08:  26 B1 A5 E3 C9 00 D0 F7
$xx10:  85 E6 A9 BE 85 E7 20 11
$xx18:  B0 20 26 B1 A5 E3 C9 0D
$xx20:  D0 F7 E6 E7 20 11 B0 18
$xx28:  60
```



8) You now have a deprotected program disk. The demo side of the disk can be copied with any disk copy program like COPYA or Disk Muncher, so make a backup of that and you're all done!

A Few Final Notes

You may have noticed that the controller presented here was a little unusual. Line 10010, which is usually used to load a foreign RWTS for Super IOB, was used instead to load FVSTUFF. This is a modification of the techinque used in the Bank Street Writer softkey in COMPUTIST No. 25 (pg.27, step 14). Here is what our controller actually does. It loads our code in at $1900 and just keeps it there while the copy is being made. (Memory from $1900 to $20FF is available as long as a swap controller is not being used). Then, after track $21 has been written, the track buffer, which normally starts at $2700, is made to start at $1900. This change is made by line 1100 of the controller. The code is then written out normally to track $22.



controller

```
1000 REM FANTAVISION CONTROLLER
1010 TK = 0 :ST = 0 :LT = 34 :CD = WR
1020 T1 = TK : GOSUB 490 : GOSUB 170 : RESTORE
1030 GOSUB 430 : GOSUB 100 :ST = ST + 1 : IF ST <
     DOS THEN 1030
1040 IF BF THEN 1060
1050 ST = 0 :TK = TK + 1 : IF TK < LT THEN 1030
1060 GOSUB 490 :TK = T1 :ST = 0 : GOSUB 230
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 POKE BUF ,25
1110 GOSUB 430 : GOSUB 100 :ST = ST + 1 : IF ST <
     2 THEN 1110
1120 HOME : PRINT "COPY^ COMPLETE" : END
1130 DATA 222 ,171 ,222 ,170
10010 PRINT CHR$ (4 ) "BLOAD^ FVSTUFF ,A$1900"
```

controller checksums

```
1000  - $356B      1080  - $E2AD
1010  - $6344      1090  - $B204
1020  - $2DB1      1100  - $08DC
1030  - $3BB0      1110  - $D1DB
1040  - $7DCB      1120  - $4CD1
1050  - $1F06      1130  - $4DD0
1060  - $334E      10010 - $7BE3
1070  - $3B4B
```

Softkey for ...

SPY VS SPY

by Danny Pollak

First Star Software, Inc.
18 East 41st Street
New York, NY 10017

Requirements:
Apple][Plus or equivalent
Spy vs Spy
Super IOB v1.5
One blank disk

Spy vs Spy is a great game in which the main objective is to escape the embassy. To escape, you must find, fill and keep the briefcase. You must then find the only exit and board the plane with the following: the passport, the traveling money, the key, and the secret plans. There are eight different levels of play and when the computer is acting as the second player, it has five different levels of intelligence. In the higher levels of play, this game can be frustratingly hard.

The Protection

An examination of the original game disk will show that tracks $00 through $11 are formatted normally. Tracks $12 through $22 appear to contain no valid data. If you were to copy tracks $00 through $11 and then attempt to play the game, the program would load in, but would bomb when it was time to play the game.

By boot code tracing the disk, I was able to determine the location of the routine which was causing this to happen and bypass it. Before this routine is called, it is read from the disk, decoded, and checksummed. If the checksum fails, memory is erased.

Next, the disk drive head is brought to track $01 and the program jumps to the routine just loaded. Track one was written at a certain speed, with sixteen sync bytes preceding the address header and four sync bytes preceding the data header. This is important to the count performed here. The nibble count routine reads from the disk until it finds the series of bytes $FE DE (part of the address header from track $01 sector $01). Then, while reading the disk, it "counts" the number of $FF's it encounters until it again finds the series of bytes $FE DE. If it doesn't like what it finds, memory is cleared and you have to reboot.

The Procedure

1) Load Super IOB and install the controller below. Follow the prompts, answering YES when asked if you wish to initialize the duplicate disk.

2) When Super IOB is finished, you will have a deprotected version of Spy vs Spy. The following sector edit was performed on the copy to defeat the nibble count routine.

Track	Sector	Byte	From	To
$0B	$08	$31	$4C	$60

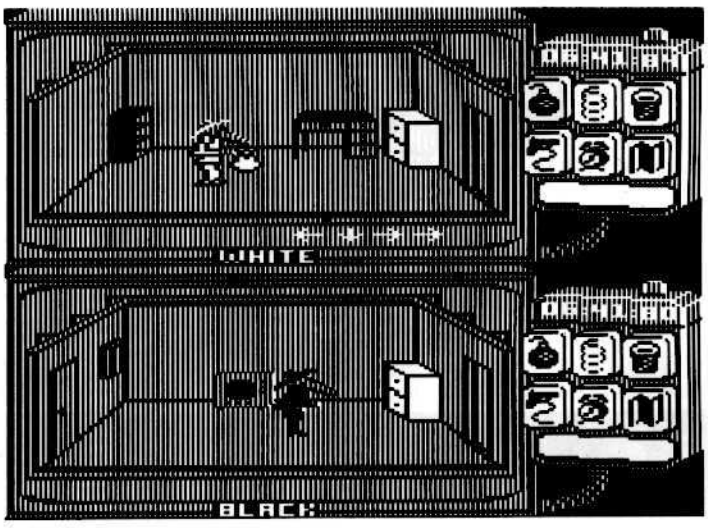
3) Put your original away and play to your heart's content. Use your traps wisely, and remember where you put them!

controller

```
1000 REM SPY VS SPY
1010 TK=0:LT=18:ST=15:LS=15:CD=WR:FAST
     =1
1020 GOSUB 490 : GOSUB 610
1025 IF TK=7 THEN T1=TK:TK=PEEK(TRK):GOSUB
     310:TK=T1
1030 GOSUB 490 : GOSUB 610 : IF PEEK(TRK)=LT
     THEN 1050
1040 TK=PEEK(TRK):ST=PEEK(SCT):GOTO 1020
1050 HOME : PRINT "COPYDONE" : END
5000 DATA 1^ CHANGES
5010 DATA 11,8,49,96
```

controller checksums

```
1000 - $356B      1040 - $B2D3
1010 - $71CC      1050 - $6CB4
1020 - $8099      5000 - $BEDA
1025 - $CFDF      5010 - $E7C3
1030 - $49CF
```



Increasing Your Disk Capacity

by Phil Goetz

Requirements:
A blank disk
A sector editor (optional)

There are two simple ways to increase the capacity of your disks. One is to use sectors on track 2 unused by DOS, and the other is to use tracks beyond track 34. (There are more, but they involve deleting DOS and stealing catalog sectors).

Many articles have been written on how to gain 11 more sectors from track 2. Actually, you can claim 15 of the 16 sectors on track 2. Eleven of these sectors are unused by DOS, two are used for $B400-B5FF, which don't need to be loaded on bootup (they contain information that is created from scratch anyway), and two sectors on track 0 (sectors $A and $B) that normally contain a DOS relocator in a master disk are left empty in a slave disk.

To use these 15 sectors, we must:

(a) Move all the sectors from track 0, sector $C through track 2, sector $A backward two sectors so they reside from track 0, sector $A through track 2, sector 0.

(b) Prevent DOS from loading or writing $B400-B5FF and make it think the original track 0, sectors $A and $B never existed.

(c) Have new disks identify track 2, sectors 1-$F as free for use.

Here are the steps to do that.

1) Boot a normal DOS 3.3 disk.

2) To "move" the range of sectors from track 0, sector $C to track 2, sector 2, we don't actually move anything. We simply tell DOS to write what normally goes on those sectors to different sectors.

To free up the sectors normally used for $B400-B5FF, we tell DOS to skip over them when loading and writing DOS. The routine for loading these sectors is at $B700-B749; the routine for writing DOS is at $B74A-B78C.

Type in the following.

```
CALL-151
B71A:00
B71E:A0 B3 EA EA
B7E0:17
B74A:A9 B4 EA
B754:A9 B3 8D F1 B7 EA
EA EA EA
B763:00
```

3) To have new disks mark these sectors as free, we need a simple patch to the initialization routine. Let's put the patch at the end of the $B6B3-B6FC area.

```
B6F5:A9 FE 8D FC B3 4C
FB AF
AEB3:08
AECD:F5 B6
```

But wait, there's more...

The second means of increasing capacity, using extra tracks, can be done with almost all disk drives. Mine is from 1980 and it can reach track 35. Some non-Apple drives can reach track 39. To use 36 tracks, we must:

(d) Inform DOS that it now has 36 tracks per disk.

(e) Have DOS initialize track 35 along with the others.

(f) Have DOS mark track 35 as being free.

Each of these modifications requires one byte. Continuing after freeing the sectors on track 2:

4) Get a new disk and make sure it is blank by CATALOGing it. It is important to do this now, because if you were to CATALOG between steps 5 and 6 you would erase the new value you will have put in $B3EF.

5) Type in the following.

```
B3EF:24
BEFE:24
AEB5:90
```

6) INITialize the blank disk. It will have 527 free sectors if you delete the Hello program, instead of 496 as normal. Fifteen are from track 2 and sixteen are from track 35.

7) Run a sector editor such as DiskEdit or the Inspector. If you can write different things to tracks 34 and 35 ($22 and $23) and read them back reliably, you can use track 35. If you cannot, repeat the instructions starting with step 1 but skipping step 5 so that you can at least gain 15 sectors.

If your disk drive can use more tracks (up to 40), add 1 to the values at $B3EF and $BEFE and 4 to the value at $AEB5 for each extra track. Using 40 tracks will give you 591 free sectors per disk. Note that drives which allow you to use 70 tracks or so, do this by writing on half-tracks as well as normal tracks. Thus, don't expect your 70 track drive to use 70 tracks by this method.

softkey for...

Dragonworld

by Timothy James Strelchun

Telarium Corp.
1 Kendall Square
Cambridge, MA 02139

Requirements:
Apple][,][Plus or //e with 64K
Super IOB
Five blank disk sides
(2-1/2 double sided disks)
Dragonworld

Dragonworld is a hi-res text (both upper and lower case) adventure game where you try to save the Last Dragon from a kidnapper and death. Along with the text, the game displays pictures showing your location throughout the game world (this is a step above the plain boring text adventures).

After deciding to back up Dragonworld, I recalled that during the bootup an Applesoft prompt was displayed, indicating to me that the game must use a fairly normal DOS, and that with a little more investigating I could make a backup copy with Super IOB.

The investigation proved to be quite revealing. I discovered that all the tracks on Disk A had abnormal address field epilogue bytes (normally DE AA). Further snooping showed that the even tracks (0, 2, 4, etc.) had normal address headers and that the odd tracks (1, 3, 5, etc.) had changed address headers (instead of D5 AA 96, it was now D4 AA 96). The other four disks looked fairly normal. Now all I had to do was write a controller for Super IOB. I used the standard DOS controller and made several changes to it to account for the protection on Disk A (some changes made were to ignore address field epilogue bytes, and to set the first byte of the address field to D4 on odd tracks). After having used Super IOB, I booted up SIDE A (the copy I just made) and surprisingly the protected DOS operated fine with the normal address field headers and epilogue bytes. I think (although I did not look through the entire DOS listing) this was achieved by not checking the changed data in the DOS routines.

Making it COPYAable

1) Format five blank disk sides with an empty Hello file. (Do the following five times:)

```
FP
INIT HELLO
```

2) Install the Dragonworld controller, listed at the end of this article, into Super IOB by your favorite method.

3) RUN Super IOB and use it to backup Side A of Dragonworld.

4) Now RUN COPYA to backup Sides B-E.

5) Be sure to put a write protect tab on all the sides (A-E).

controller

```
1000 REM DRAGONWORLD CONTROLLER
1010 TK = 0 :ST = 0 :LT = 35 :CD = WR
1020 POKE 47507 ,0 : POKE 47517 ,0
1030 T1 = TK : GOSUB 490
1040 POKE 47445 ,213
1050 IF TK < > ( INT (TK / 2 ) ) * 2 THEN POKE
     47445 ,212
1060 GOSUB 430 : GOSUB 100 :ST = ST + 1 : IF ST <
     DOS THEN 1060
1070 IF BF THEN 1090
1080 ST = 0 :TK = TK + 1 : IF TK < LT THEN 1040
1090 GOSUB 490 : POKE 47445 ,213 :TK = T1 :ST = 0
1100 GOSUB 430 : GOSUB 100 :ST = ST + 1 : IF ST <
     DOS THEN 1100
1110 ST = 0 :TK = TK + 1 : IF BF = 0 AND TK < LT THEN
     1100
1120 IF TK < LT THEN 1030
1130 POKE 47507 ,174 : POKE 47517 ,164 : POKE
     47445 ,213
1140 HOME : PRINT : PRINT "DONE^WITH^SIDE^A."
     : END
```

controller checksums

```
1000  - $356B      1080  - $0507
1010  - $3266      1090  - $A256
1020  - $5917      1100  - $9356
1030  - $0A12      1110  - $F70D
1040  - $AFB3      1120  - $9811
1050  - $26A1      1130  - $BC98
1060  - $29A5      1140  - $D731
1070  - $077B
```

Adventure Tips

SECRET AGENT MISSION #1

- Forks keep nurses away!
- Registers have money inside them.
- Can't get a tie, try stealing one.
- Hairpins are good lock picks.
- You don't just get drinks in a bar.
- Climbing up ladders will give you an extra step!
- Secret agents like you should practice climbing through windows.
- Where there are registers with money, there are I.D. cards upstairs!

By Jesse Weissman

MYSTERY FUN HOUSE

- The trampoline is a portable room for storage.
- The "OUT OF ORDER" sign may be put on something to prevent its use if such is an inconvenience to you.
- The mermaid will appreciate your combing her hair.
- The gum tastes awful,- because it's not gum, but something you'll need for a REAL bang.

By Paul Wilson

CRYPT OF MEDEA

- Examine the head and search the body.
- Examine the moss.
- If no results occur in one room, try the next.
- Turn off the graphics mode, before lighting the bomb fuse and putting it down.
- The gloves are usable more than once. So is the shovel.
- The mask will help you to breathe in one place.

By Paul Wilson

HITCH HIKERS GUIDE

- Put the towel on your head anywhere.
- Real tea will make you real happy.
- You can't afford to retain common sense, once you have the chance to get rid of it.

By Paul Wilson

SHERLOCK HOLMES

- Do not jump overboard.
- Do not try to kill Dr. Watson.

By RAK

FreeWare

PUBLIC DOMAIN SOFTWARE FOR YOUR APPLE

What is Public Domain Software?

Public Domain Software (PDS) consists of programs that are donated to the public, and therefore, have no copyrights attached. They are written by a variety of people, some professionals, some not—in most cases each program is NOT commercial-quality and is not supported as such.

Who can use the Library?

Our library is supplied in DOS 3.3, 16 sector format for Apple II computers. Please note that a few machine language programs will function erratically on the IIe and IIc because of changes in the F8 monitor. We have not tested all of the programs nor do we have a list of what works. So be careful—"Caveat Emptor."

What does the Computer Learning Center do?

The Computer Learning Center provides a service that copies and DISTRIBUTES software in the public domain. Our library is constructed on a "per volume" basis, each volume containing approximately 20 programs. The $4/volume fee covers the cost of the disk and costs involved in copying, labeling, packaging, mailing and other related expenses. Due to the nature of our library, PDS cannot be returned for a cash refund or exchanged for different volumes.

When using our PDS listings, the volume name and number is in **reverse** print. Use the volume number next to the name when you order. Each volume name is followed by a list of programs on that disk. (Except Eamon, where only the scenario title is listed.) The left column of this list indicates the language required by the program. (A—Applesoft, I—Integer, B—Machine Code, and T—Text Files.)

How to Order:

1. Select the volumes with the programs that you want.
2. Check the numbered boxes on this form that match your selections.
3. For every 10 volumes that you order, you get 1 free bonus volume. Circle your FREE bonus selection on the order form.
4. Fill in the address information. (Please print neatly.)
5. Total the number of volumes that you are ordering and multiply this number by $4.00. The minimum order is two (2) volumes.

- *Washington residents add 7.8% sales tax.*
- *Overseas, Canada, and Mexico: add 20% for shipping.*

6. Send a check or money order for the total amount due. We accept VISA/MC. Credit card orders must have a valid signature. We accept international money orders (in USA funds) and checks drawn on USA banks. Canadian checks must specify USA dollars. Make them payable to: **Computer Learning Center P.O. Box 110876-HC Tacoma, WA 98411**

☐ E01	☐ E10	☐ E19	☐ E28	☐ E37	☐ E46	☐ E55	☐ E64	☐ E73	☐ E82
☐ E02	☐ E11	☐ E20	☐ E29	☐ E38	☐ E47	☐ E56	☐ E65	☐ E74	☐ E83
☐ E03	☐ E12	☐ E21	☐ E30	☐ E39	☐ E48	☐ E57	☐ E66	☐ E75	☐ E84
☐ E04	☐ E13	☐ E22	☐ E31	☐ E40	☐ E49	☐ E58	☐ E67	☐ E76	
☐ E05	☐ E14	☐ E23	☐ E32	☐ E41	☐ E50	☐ E59	☐ E68	☐ E77	
☐ E06	☐ E15	☐ E24	☐ E33	☐ E42	☐ E51	☐ E60	☐ E69	☐ E78	
☐ E07	☐ E16	☐ E25	☐ E34	☐ E43	☐ E52	☐ E61	☐ E70	☐ E79	
☐ E08	☐ E17	☐ E26	☐ E35	☐ E44	☐ E53	☐ E62	☐ E71	☐ E80	
☐ E09	☐ E18	☐ E27	☐ E36	☐ E45	☐ E54	☐ E63	☐ E72	☐ E81	

☐ P01	☐ P02	☐ P11	☐ P18	☐ P19	☐ P20	☐ P21	☐ P36	☐ P50	☐ P59
☐ P61	☐ P67	☐ P71	☐ P76						

Send me the volumes that I have checked. I understand that the minimum order is two volumes.

Name_____

Address_____

City _____ State _____ Zip _____

Country _____ Phone _____

VISA MC _____ - _____ - _____ - _____ Exp _____

Signature _____

Public Domain Software is supplied **as-is**. HC 4/86-30

What is an Eamon Adventure?

Eamon Adventures are a collection of entertaining, flexible, fantasy role-playing, **text-adventures** in the public domain.

MASTER/Beginner's Cave.

The Master Diskette is **required** to play EAMON. It is used to create your character and stores the character data between adventures. This volume also contains the Beginner's Cave—a short (but not entirely safe) romp for your new character. (Highly recommended training for new adventurers.)

EAMON Adventures

E1 MASTER/Beginner's Cave	E43 Tomb of Y'Golonac
E2 Lair of the Minotaur	E44 Operation Crab Key
E3 Cave of the Mind	E45 Feast of Carroll
E4 Zyphur River Venture	E46 The Master's Dungeon
E5 Castle of Doom	E47 Crystal Mountain
E6 Death Star	E48 Lost Adventure
E7 Devil's Tomb	E49 The Manxome Foe
E8 Abductor's Quarters	E50 Behind the Sealed Door
E9 Assault of the Clone Master	E51 Land of Death
E10 Magic Kingdom	E52 Jungles of Vietnam
E11 Tomb of Molinar	E53 Black Castle of Nagog
E12 Quest for Trezore	E54 Sewers of Chicago
E13 Caves of Treasure Island	E55 Caverns of Doom
E14 Furioso	E56 Valkenburg Castle
E15 The Heroes' Castle	E57 Modern Problems
E16 Caves of Mondamen	E58 Priests of Xim
E17 Merlin's Castle	E59 Escape from the Orc Lair
E18 Hogarth Castle	E60 Castle of Count Fuey
E19 Death Trap	E61 Search for the Key
E20 The Black Death	E62 The Rescue Mission
E21 Quest for Marron	E63 The Maze of Quasequeton
E22 Senators' Chambers	E64 Chamber of the Dragons
E23 Temple of Ngurct	E65 Swordquest
E24 Black Mountain	E66 Smith's Stronghold
E25 Nuclear Nightmare	E67 Picnic in Paradise
E26 Assault on the Moleman	E68 The Caves of Eamon Bluff
E27 Revenge of the Moleman	E69 Future Quest
E28 Tower of London	E70 Castle Kophinos
E29 Lost Island of Apple	E71 The Devils Dungeon
E30 Underground City	E72 Harpy Cloud
E31 Gauntlet	E73 The School of Death
E32 House of Ill Repute (Adult)	E74 The Dungeons of Xenon
E33 Orb of Polaris	E75 Chaosium Caves
E34 Death's Gateway	E76 Life Quest
E35 Lair of the Mutants	E77 Dharmaquest
E36 Citadel of Blood	E78 Mean Streets
E37 Quest for the Holy Grail	E79 The Temple of the Guild
E38 City in the Clouds	E80 Deep Canyon
E39 Museum of Unnatural History	E81 The Castle of Ravineta
E40 Deamons Playground	E82 The Prince's Tavern
E41 Caverns of Lanst	E83 The Search for Yourself
E42 Alternate Beginner's Cave	E84 The Temple of the Trolls

FreeWare: MORE Public Domain Software

P01 Apple Tutor

I 036 Basic Programming 1
I 047 Basic Programming 2
I 048 Basic Programming 3
I 044 Basic Programming 4
A 004 Basic-Integer
B 022 Basic-Integer.X
I 019 Conventions
I 016 CPU 6502
I 002 Hello Sample
I 030 Micro 6502 Simulation
I 051 Mini Assembler Tutorial
A 022 Random Drill Tutor
I 007 Sweet 16 Disassembler
I 004 Sweet 16 Speed ?
I 026 Top Down Programming

P02 Apple Tutor

A 021 Apple II Demo
A 004 Basic Exercise
I 043 Basic-Applesoft
I 093 Disk Aide.13
I 038 Disk Aide DOC
B 003 Disk Aide.X
A 028 DOS System Instruction
A 013 Pilot Version I
I 054 Program Devel. Pkg
A 014 Suppl. Mini Assembler
B 006 Supplement.X
I 003 Text File Read
I 003 Text File Write
I 009 Text Hello
T 009 Text How To
T 010 Text Intro
T 006 Text Peek Poke Call
T 010 Text Programming
T 007 Text Redbook
T 011 Text Software
A 002 Buzz

P11 Art & Graphic

A 006 Art Align
A 002 Art Bars
A 003 Art Circle
A 002 Art Demo
A 003 Art Diverging Circles
A 003 Art Diverging Octagons
A 003 Art Double Cross
A 006 Art Dougs Theme
A 003 Art Drifting Circle
A 003 Art Drifting Octagon
A 004 Art Elephant
A 003 Art Figure 8
A 005 Art Horizon
A 007 Art IBM
A 003 Art Octal 8
A 002 Art Oneliner 7
A 007 Art Super Kalied
A 003 Art Twist II
A 015 Art Xmas Card
A 014 Auto Space War II
A 002 Billboard Mother
A 002 Graphic Switch
A 010 Graphics Tablet
A 008 Hi-res Text Demo ?
B 007 Hi-res Text Set
A 004 Higher Hires
A 006 Hires Sketch
A 003 Invert Monitor Mode
A 035 Life
B 015 Life Language
A 004 Life Leader
A 002 Picture Loader
A 016 Plot Pourri
A 010 Poster ?
A 008 Poster Banner I
A 009 Poster Love II
A 003 Rubber Apple
A 020 Shape Editor
A 010 Shape Generator
A 003 Shape Instr
A 023 Shape Table Editor
T 001 Shape Table for Hires Label
T 002 Shape Test
A 015 Skywriter
A 003 Skywriter I
A 003 Skywriter II
A 025 Skywriter Instr
A 013 Skywriter Snoopy
B 018 Star.shape
A 005 TV Pattern Generator

P18 Business & Finance

A 005 Annuity Principal & Int.
A 003 Annuity Reg. Deposits
A 018 Annuity ◄
A 013 Bond Price & Interest
A 012 Bond Value
A 032 Budget Monthly
A 013 Decision Matrix
A 054 Financial Pak
A 016 Invest. Annuity Forecast
A 012 Keogh Savings Program
A 014 Loan Amort. Schedule
A 008 Loan Direct Reduction
A 004 Loan Interest
A 029 Market Evaluator Pak
A 009 Mortgage Calculation
A 007 Nicer Writer ◄
A 003 Regular Deposits I
A 007 Sales Tax At 6%
A 017 Security Analysis
A 006 Sec. Analysis Copy Data
A 003 Simple Interest
A 010 Stock Option Analysis
A 016 Stock Op covered hedge
A 015 Stock Option Pricing I
A 021 Stock Option Pricing II
A 008 Stock Portfolio Valuation
A 028 Stocks
A 010 Trip Cost Analysis

P19 Business & Finance

I 030 Calendar Personal
I 006 Letter Writer
I 006 Letter Writer Enhance
I 014 Phone List
I 005 Real Estate Plot
B 034 Real Estate Plot.X
A 012 Stock Monitor I
A 015 Stock Monitor II
A 024 Trend Line Analysis
A 006 Visicalc Coord Formulas
A 012 Visicalc D File Printer
A 014 Visicalc Formulas
A 006 Visicalc Formulas Instr

P20 Business & Finance

A 025 Apartment Mortgage
A 052 Banking And Finance
A 058 Business Finance
A 006 Check Stub
A 020 Household Exp. Profile
A 041 Income Tax 1040 For 77
A 029 Inventory Company
A 004 Inventory Cost File Entry
A 016 Inventory Home I
A 022 Inventory Model
A 009 Inventory Print ◄
A 003 Inventory Shortest
A 003 Inventory Shortest Read
A 023 Inventory
A 003 Inventory.DOC
A 019 Inventory.File Create
A 029 Inventory.File Read
A 018 Stock Market Forecaster
A 011 Stock Valuation

P21 Business & Finance

A 004 Annuity
A 004 Average Growth Rate
T 001 Basenamefile
A 046 CAC Record System ◄
A 009 Check Book Balancer
A 008 Check Writer
A 004 Depreciation Amount
A 003 Depreciation Rate
A 010 Depreciation Schedules
A 003 Depreciation Straight
A 004 Disc. Commercial Paper
A 024 House Sales
A 005 Income Taxes
A 012 Interest Earned
A 003 Interest Rate
A 038 Inventory Home II
A 004 Invest. For Withdrawals
A 004 Investment Future Value
A 004 Investment Initial
A 013 Lease Computation
A 031 Life Mgmt And Finances
A 012 Life Mgmt Txt Organizer
A 005 Loan Balance
A 007 Loan Interest Rate
A 005 Loan Last Payment
A 004 Loan Principal
A 005 Loan Regular Payment
A 004 Loan Term
A 004 Mortgage Computer
A 007 Mortgage Table
A 063 Payroll
A 004 Regular Deposits II
A 003 Regular Withdrawals
A 004 Salvage Value
A 009 Savings Growth
A 011 Survey Data Reduction
A 003 System Reliability
A 004 Treasury Bill Valuation

P36 Game

A 027 Blackjack Strategy
A 012 Combat
A 017 Craps BW
A 035 Cribbage I BW
A 028 Football Predictions
A 011 Fox And Hounds
A 026 French Military Game
A 020 Gold Mine
A 022 Golf II
A 016 Hi Q
A 027 Hockey I
A 021 Horse Race III
A 031 Kingdom
A 008 Literature Quiz
A 026 Marooned In Space
A 004 Ping Pong
A 010 Robot BW
A 016 Survive
A 018 Twonky I
A 018 Word Maze Maker
A 003 Football Predictions.note

P50 Game

A 017 Battle
A 015 Century 51
B 002 Century 51.X
A 020 Checkers II
A 021 Commodity Market
A 016 Craps I
A 013 Cryptograms
A 017 Frustration
A 027 Geography II
A 015 Hangman II
A 022 Lunar Landing BW
A 061 Market Crash
B 005 Market Crash.X
A 009 Name the States
A 018 Puzzle Generator
A 007 Solitaire Checker Puzzle
A 010 Spell the States
A 012 Subscan for Two
B 002 Subscan.X
A 031 World War

P59 Math & Statistics

A 008 Anglo To Metric I
A 011 Anglo To Metric II
A 003 Arcsin Arccos
A 016 Calculator
A 007 Calculus I
A 013 Calculus II
A 007 Cash Register
A 005 Circle Area Circum
A 009 Compound Interest
A 010 Critical Path Analysis
A 006 Curve Fit I
A 013 Curve Fit II
A 013 Curve Fit III
A 003 Derivative of Equation
A 010 Differential Eqn Solver
A 015 Equation Solver
A 005 Equations I
A 014 Equations II
A 014 Factor
A 003 Fibonacci Numbers
A 012 Foot Candle Analysis
A 010 Fourier
A 011 Fourier Transform
A 007 Gaussian Quadrature II
A 010 Math Drill I
A 011 Math Drill II
A 005 Matrix Inversion II
A 009 Matrix Operation
A 016 Mean Vari Stdr Devia II
A 007 Mode Finder
A 003 N Factorial
A 007 Plot Consec. Reactions
A 006 Plot Functions of X
A 003 Prog Chart ?
A 011 Robot Motor Design
A 005 Sine Function
A 010 Statistics I
A 014 Statistics II
A 014 T Test Stdn Deviation
A 004 Time Speed Dist. Exer.
A 006 Time Speed Dist. Fuel
A 014 Triangle Solver
A 011 Unpaired Group Comp.

P61 Math & Statistics

A 003 Angle Conversion
A 012 Anglo To Metric III
A 003 Area of Polygon
A 008 Binomial Distribution
A 004 Blackbody
A 003 Chi Square Distribution
A 004 Chi Square Test
A 004 Coordinate Conversion
A 008 Coordinate Plot
A 003 Curvilinear Interpolation
A 002 Derivative
A 016 Dfit
A 011 Differential Eqn Solver
A 004 Exponential Regression
A 004 F Distribution
A 004 Gaussian Quadrature I
A 003 Geometric Mean
A 004 Geometric Regression
A 004 Greatest Comn Denom.
A 005 Histogram
A 003 Linear Interpolation
A 007 Linear Programming
A 004 Linear Regression
A 005 Mann Whitney U Test
A 022 Math Drill III
A 019 Math Multiply Drill
A 019 Math ◄
A 013 Matrices
A 004 Matrix Inversion I
A 004 Matrix Multiplication
A 004 Matrix Operation Simple
A 004 Mean Vari Stdr Devia I
A 007 Mult. Linear Regression
A 004 Normal Distribution
A 006 Nth Order Regression
A 004 Number Combinations
A 002 Parabola Plot
A 003 Permutation Comb. I
A 007 Permutation Comb. II
A 003 Poisson Distribution
A 006 Polar Equation Plot
A 011 Polyfit
A 013 Polynomial Regression
A 006 Power Curve Fit ©
A 003 Prime Factors I
A 006 Prime Factors II
A 003 Quadratic Formula
A 006 Quadratic Surface
A 015 Right Triangle Solver
A 016 Root Finder
A 005 Roots of Poly Half
A 005 Roots of Polynomials
A 004 Simpson's Rule
A 004 Simultaneous Equations
A 004 T Distribution
A 005 T Distribution Test
A 003 Trapezoidal Rule
A 005 Triangle Factors
A 007 Triangle Parts
A 003 Trig Polynomial
A 004 Vector Analysis
A 003 Vector Operations

P67 Passion (Adult)

A 002 Form 0 Startup
B 034 Form 1
B 034 Form 2
B 034 Form 3
B 034 Form 4
B 034 Form 5
B 034 Form 6
B 034 Form 7
B 034 Form 8
B 034 Form 9
A 005 Hot Apples I
A 006 Hot Apples II
A 013 Touch I
A 013 Touch II
I 013 Zoom 1
I 013 Zoom 2

P71 Pastime & Other

A 048 Bio-Rhythms
A 011 Calendar One Month
A 007 Calendar Perpetual
A 020 Calendar Reminder
I 020 Colossus
A 011 Decision Maker III
I 063 Dirty Tricks Beware
A 025 Dirty Tricks II
B 018 Dirty Tricks.X
I 020 Horoscope
A 026 Miles Per Gallon Record
I 074 Numerology
I 002 Phone Mnemonic 1
B 003 Phone Mnemonic 1.X
A 010 Probability
A 021 Random Insults
I 002 Random Words 1
I 003 Syn Tax
I 008 Timer

P76 Utility

A 003 Base Conversion Chart I
A 003 Base Conversion Chart II
A 006 Base Convert **
A 005 Base Convert Beymer
A 012 Base Convert Ellmers
A 005 Base Convert Jenkins
A 029 Base Convert Massimo
I 043 Basic-Applesoft
A 007 Binary To FP
A 053 Calendar And Posters
A 002 Capture A Program
A 005 Catalog Printer
A 003 Clear Memory
A 004 Copy Text File I
A 004 Copy Text File II
A 004 Copy Text File III
A 004 Ctrl Char Catalog
A 003 Ctrl Char Reveal
A 003 Dump ASCII Memory
A 002 Erase Hires Screen
A 011 Format #
A 003 Free Sectors Aldrich
A 004 Free Sectors Brown
B 002 Free Sectors Brown.X
A 019 Illegal Commands
A 005 List Page Diaz
A 005 List Page Wysocki
B 002 List Page Wysocki.X
B 002 List Page.X
A 007 Mach To Pokes Conv
A 002 Musical Keys
B 002 Musical Keys.X
A 013 Phone Dialer
A 003 Pointers
A 002 PR# 6 On Reset
A 003 Random Sentence
B 002 Rem Stripper
A 003 Rem Stripper Doc
A 006 Renumber
A 004 Renumber Merge
A 003 Reverse Print
A 018 Sort Catalog
A 015 SSM AIO Board
A 004 String Comparator
A 004 String In A String Search
A 004 Text File Edit
A 003 Text File Peek
A 003 Text File Read & Print I
A 005 Text File Read & Print II
A 005 Text File Write
A 004 Token Address Table I
A 005 Utility Statements

Back Issues

of COMPUTIST *(formerly Hardcore COMPUTIST)*

are still available, though some issues (marked NA) are sold out, Library disks are available for ALL issues of COMPUTIST and even the old COREs.

Don't

in programs that appear in COMPUTIST.

Order the Library Disk, instead!

For each issue of COMPUTIST, a Library Disk containing all the programs that appeared in that issue is prepared for **SMART READERS** like you who have *better* things to do with their time than type in program listings. Please use the order form to order either the magazines or library disks or BOTH MAGAZINE AND DISK AND SAVE $1.75. Documentation for Library Disks is in the corresponding issue.

Back Issues and Library Disk order form

Issue	Mag $4.75	Disk $9.95	Both $12.95
30	☐	☐	☐
29	☐	☐	☐
28	☐	☐	☐
27	☐	☐	☐
26	☐	☐	☐
25	☐	☐	☐
24	☐	☐	☐
23	☐	☐	☐
22	☐	☐	☐
21	☐	☐	☐
20	☐	☐	☐
19	☐	☐	☐
18	☐	☐	☐
17	☐	☐	☐
16	☐	☐	☐
15	NA	☐	NA
14	NA	☐	NA
13	☐	☐	☐
12	NA	☐	NA
11	NA	☐	NA
10	NA	☐	NA
9	NA	☐	NA
8	NA	☐	NA
7	☐	☐	☐
6	NA	☐	NA
4	☐	☐	☐
3	NA	☐	NA
Core 2	☐	☐	☐
2	NA	☐	NA
1	☐	☐	☐
Core 1	☐	☐	NA
Core 3	☐	☐	☐

Special "Both" disk & magazine combination orders apply to an issue and its corresponding disk. Some disks apply to more than one issue and are shown as taller boxes.

Send me the back issues and/or library disks indicated above:

Name _____ ID# _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

VISA MC _____-_____-_____-_____ Exp. _____

Signature _____ CP30

Send check or money order to:

COMPUTIST
PO Box 110846-T
Tacoma, WA 98411

Most orders shipped UPS.
Please use street address.
In Washington state: add 7.8% sales tax.
Offer good while supply lasts.

Foreign orders please use chart below.

Back Issue Rates For Foreign Orders (effective immediately)

Magazine orders	Canada/Mexico	Other Foreign
Quantity	Price per copy	Price per copy
1 - 2	$8.00	$14.25
3 - 4	$7.00	$13.25
5 or more	$6.00	$12.25

FOREIGN DISK ORDERS

Foreign disk orders add 20% shipping.
Special "Both" disk and magazine combinations shown above do not apply to Foreign orders.
US funds drawn on US banks.
All foreign orders sent AIR RATES.

Description of Available Back Issues

29 *Softkeys* | Threshold | Checkers v2.1 | Microtype | Gen. & Organic Chemistry Series | Uptown Trivia | Murder by the Dozen | *Features* | Customizing the Monitor by Adding 65C02 Disassembly | *Core* | The Animator |

28 *Softkeys* | Ultima IV | Robot Odyssey | Rendezvous | Word Attack & Classmate | Three from Mindscape | Alphabetic Keyboarding | Hacker | Disk Director | Lode Runner | MIDI/4 | *Feature* | Capturing the Hidden Archon Editor | *Core* | Fingerprint Plus: A Review | Beneath Beyond Castle Wolfenstein (part 2) | .....

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

26 *Softkeys* | Cannonball Blitz | Instant Recall | Gessler Spanish software | More Stickybears | *Readers' Softkeys* | Financial Cookbook | Super Zaxxon | Wizardry | Preschool Fun | Holy Grail | Inca | 128K Zaxxon | *Features* | ProEdit | *Core* | Games of 1985 part I | .....

25 *Softkeys* | DB Master 4.2 | Business Writer | Barron's Computer SAT | Take 1 | Bank Street Speller | Where In The World Is Carmen Sandiego | Bank Street Writer 128K | Word Challenge | *Readers' Softkeys* | Spy's Demise | Mind Prober | BC's Quest For Tires | Early Games | Homeword Speller | *Features* | Adding IF THEN ELSE To Applesoft | *Core* | DOS To ProDOS And Back | .....

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

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

22 *Softkeys* | Miner 2049er | Lode Runner | A2-PB1 Pinball | *Readers' Softkeys* | The Heist | Old Ironsides | Grandma's House | In Search of the Most Amazing Thing | Morloc's Tower | Marauder | Sargon III | *Features* | Customized Drive Speed Control | Super IOB version 1.5 | *Core* | The Macro System | .....

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

20 *Softkeys* | Sargon III | Wizardry: Proving Grounds of the Mad Overlord and Knight of Diamonds | *Reader' Softkeys* | The Report Card V1.1 | Kidwriter | *Feature* | Apple][Boot ROM Disassembly | *core* | The Graphic Grabber v3.0 | Copy II+ 5.0: A Review | The Know-Drive: A Hardware Evaluation | An Improved BASIC/Binary Combo | .....

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

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

17 *Softkeys* | The Print Shop | Crossword Magic | The Standing Stones | Beer Run | Skyfox | and Random House Disks | *Features* | A Tutorial For Disk Inspection and the Use Of Super IOB | S-C Macro Assembler Directives (reprint) *Core* | The Graphic Grabber For The Print Shop | The Lone Catalog Arranger Part 2.....

16 *Readers' Softkeys* | Rescue Raiders | Sheila | Basic Building Blocks | Artsci Programs | Crossfire | *Softkeys* | Sensible Speller for ProDOS | Sideways | *Feature* | Secret Weapon: RAMcard | *Core* | The Controller Writer | A Fix For The Beyond Castle Wolfenstein Softkey | The Lone Catalog Arranger Part 1.....

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

7 *Softkeys* | Zaxxon | Mask of the Sun | Crush | Crumble & Chomp | Snake Byte | DB Master | & Mouskattack | *Features* | Making Liberated Backups That Retain Their Copy Protection | S-C Assembler: Review | Disk Directory Designer | *Core* | COREfiler: Part 1 | Upper & Lower Case Output for Zork.....

4 Ultima II Character Editor | *Softkeys* | Ultima II | Witness | Prisoner II | Pest Patrol | Adventure Tips for Ultima II & III | Copy II Plus PARMS Update.....

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

CORE 3 **Games:** Constructing Your Own Joystick | Compiling Games | *GAME REVIEWS:* Over 30 of the latest and best | Pick Of The Pack: All-time TOP 20 games | Destructive Forces | EAMON | Graphics Magician and GraFORTH | and Dragon Dungeon |
.....

CORE 2 **Utilites:** Dynamic Menu | High Res: Scroll Demo | GOTO Label: Replace | Line Find | Quick Copy: Copy |
.....

CORE 1 **Graphics:** Memory Map | Text Graphics: Marquee | Boxes | Jagged Scroller | Low Res: Color Character Chart | High Res: Screen Cruncher | The UFO Factory | Color | Vector Graphics:Shimmering Shapes | A Shape Table Mini-Editor | Block Graphics: Arcade Quality Graphics for BASIC Programmers | Animation |
.....

Back issues not listed are no longer available.

But disks are still available for ALL sold-out issues !

Use the order form on the other side of this page

Writer's Guide

COMPUTIST

is a monthly magazine dedicated to the serious user of the Apple (or compatible) computer. COMPUTIST welcomes articles on a variety of subjects in all levels of technical difficulty but requires accurate data, technical competence, correct English usage, readable style, and fully defined jargon and buzzwords.

MANUSCRIPT MECHANICS

All manuscripts must be typed or printed on one side of the paper. Text should be double-spaced.

Printouts should use a non-compressed font with both upper and lower case. A letter quality mode is preferred, with each page torn at the perforation only. Pages need not be stapled together. The cover page of each manuscript should contain the following data:

TITLE OF WORK
FULL NAME OF AUTHOR
ADDRESS
PHONE NUMBER

Each page of the manuscript and program listing should include the author's name, the title of the work, and the page number in the upper right hand corner.

The article and any accompanying program **SHOULD BE SUBMITTED AS A STANDARD TEXT FILE ON A DOS 3.3 DISK.** Label the disk with the title of the work and the author's full name and address. **ON DISK, TEXT MUST BE SINGLE-SPACED ONLY.** Please identify your editing program.

Original disks are always returned as soon as possible. Other materials will be returned only when adequate return packaging and postage is enclosed. We are not responsible for unreturned submissions. We *will guarantee* the return of original commercial disks mailed to us for verification of an accompanying softkey.

You will be notified of the status of your submission within 4 to 6 weeks after it is received if the article is a softkey accompanied by an original disk. Please submit completed manuscripts directly; do not query first. Previously published material and simultaneous submissions are not accepted.

SUBJECTS

We prefer material on these topics:

1) Original program/article combinations
2) General articles (Apple computing)
3) Softkeys
4) Advanced Playing Techniques (APT's)
5) Hardware modifications
6) DOS modifications
7) Product reviews (hardware and software)
8) Utilities
9) Bit Copy Parameters

WRITING YOUR ARTICLE

Observe the following points of style:

A. Always assume that your reader is a novice and explain all buzzwords and technical jargon. Pay special attention to grammar and punctuation; we require technical competence but also good, readable style.

B. Whenever appropriate, a list of hardware and software requirements should be included at the beginning of the manuscript. When published, this list will be offset from the main text.

C. Include the name and address of the manufacturer and the price when a commercial program is mentioned. This is of particular importance in PRODUCT REVIEWS.

D. When submitting programs, first introduce the purpose of the program and features of special interest. Include background information describing its use. Tips for advanced uses, program modifications, and utilities can also be included. Avoid long print statements and use TABs instead of spaces.

Remember: A beginner should be able to type the program with ease.

E. A PROGRAM is not accepted for publication without an accompanying article. These articles, as well as articles on **hardware** and **DOS modifications** MUST summarize the action of the main routines and include a fully remarked listing.

F. GENERAL ARTICLES may include advanced tips, tutorials, and explorations of a particular aspect of Apple computing.

G. SOFTKEYS of any length are acceptable and must contain detailed step-by-step procedures. For each softkey, first introduce the locking technique used and then give precise steps to unlock the copy-protected program. Number each step whenever possible. We accept articles which explain locking techniques used in several programs published by the same company.

H. When altering game programs, the changes made are sometimes extensive enough to warrant the title of ADVANCED PLAYING TECHNIQUE (APT). APTs can deal with alterations to a program, deleting annoying sounds, acquiring more points in play and avoiding hazards. Again, provide step-by-step instructions to complete each APT and explain each step's function. APT's of 100 words or more are preferred.

AUTHOR'S RIGHTS

Each article is published under the author's byline. As a rule, all rights, as well as one-time reprint rights are purchased. Purchase of exclusive rights to programs is required; however, alternate arrangements may be made with individual authors depending on the merit of the contribution.

PAYMENTS

COMPUTIST pays upon publication. Rate of payment depends on the amount of editing required and the length of the article. Payment ranges from $20 to $50 per typeset page for an article. We also pay $10 to $20 for short softkeys and APT's. A fully explained softkey accompanied by the commercial disk for verification may earn up to $50 per typeset page.

Please mail your submissions to:

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

FREE

your disks from their uncopyable status

Now you can make necessary backups of your locked-up commercial software with

The BOOK of Softkeys

shows you how to softkey (undo copy-protection on) commercial software.

Volumes II and III are being compiled now!

Volume I
(157 pages)
contains softkeys for:

Akalabeth
Ampermagic
Apple Galaxian
Aztec
Bag of Tricks
Bill Budge's Trilogy
Buzzard Bait
Cannonball Blitz
Casino
Data Reporter
Deadline
Disk Organizer II
Egbert II Communications Disk
Hard Hat Mack
Home Accountant
Homeword
Lancaster
Magic Window II
Multi-disk Catalog
Multiplan
Pest Patrol
Prisoner II
Sammy Lightfoot
Screen Writer II
Sneakers
Spy's Demise
Starcross
Suspended
Ultima II
Visifile
Visiplot
Visitrend
Witness
Wizardry
Zork I
Zork II
Zork III

plus how-to articles and program listings of need-to-have programs used to make unprotected backups.

If you want to make backups then you want The Book Of Softkeys!

For your copy of The Book Of Softkeys Volume 1, send $20 (Foreign orders add 20%. U.S. funds drawn on U.S. banks. Washington state orders add 7.8% sales tax.) to:
Softkey Publishing; PO Box 110816-T; Tacoma, WA 98411