Copy II Plus Tips, Tricks & Tech Info Collectives

Creating .DSK images WITHOUT using null-modem/ cables etc.

I've only just joined this discussion, so I'm not sure if this method has been discussed before, but I've used it for 'years'. It creates ..dsk images that work fine with AppleWin, Catakig, Bernie etc.

Necessary hardware and software etc.:

- Apple II with a 5.25 and 3.5 drive (I use a GS with the two drives daisy-chained).
- A Macintosh. A PPC with sys 7.5 or later makes it easy to read 800K Prodos disks, (and readily create MSDOS disks containing .dsk files, to take across to a PC).
- At least one 800K floppy disk.
- Copy II+ for Apple2 (I use vers. 7.3, but earlier/later versions work ok.)
- Software for formatting 800K disks (Apple2 system disk vers 3-->is fine)
- 'Easy Convert' for Mac. (Available at http://www.emulation.net/apple2/index.html)

Also, it is assumed that the 5.25' Apple disks that you intend to convert to ..dsk image format are NOT copy-protected.

Steps: (starting on the Apple2 or GS)

- Format your 800K disk (Prodos). Give the disk any name (eg. DSK-IMAGES).

- Start up Copy II+, and select COPY at the front screen, then select DISK option.
- Set the SOURCE drive to be SLOT 6, DRIVE 1 (ie. the 5.25 drive)
- Set the TARGET DEVICE to be SLOT 5, DRIVE 1 (ie. the 3.5 drive)

Copy II+ will now ask you to INSERT DISKS

- Insert the SOURCE Apple2 disk (the 5.25 ORIGINAL of the program to be converted).
- Insert the TARGET disk (the blank formatted 800K 3.5 floppy)

- Press RETURN (Copy II+ will then ask you for a TARGET FILE NAME)
- Type in a name (eg. MYDISK1) and press RETURN

Copy II+ will then proceed through a series of READs and WRITEs FROM the 5.25 original TO the 3.5 floppy in the TARGET drive, until the copy is finished.

o Take the 800K floppy across to your Macintosh
and copy the raw 'MYDISK1' file
  to the Mac's hard-drive, into a folder with the
program 'Easy Convert').

o Run Easy Convert and select the raw MYDISK1 file
(sorry no drag and drop).
  When asked for a name, just add .dsk to the
existing name and save as MYDISK1.dsk.

  Easy Convert re-orders the track sectors to be
emulator-readable, and gives the
  .dsk image the appropriate creator/type
designation of A2EM/DSK5.

You can now use these .dsk images on Mac-based
emulators (Catakig, Bernie etc.)
or put them (9 at a time) on 1.4 Mb PC-formatted
floppies for simple transport
across to Win or DOS based emulators (AppleWin
etc.) on a PC (no cables required!!).

Cheers,
Lachlan Arnott

Wings of Fury, VCR Companion, CrossWord Magic 4.0, Tetris, RoboCop, Ikari Warriors, etc. etc. I have a solution. It took many days to come across this one.

First, you need Copy II+ 6.5 or later, or the equivalent of a Bit Copier with a Nibble Editor and then some patience. Also, it is much easier if you have a Sector Editor with a string or hex search mode.

If you don't have a sector editor then skip this step. If you do, go to the mode to patch the sector editor. Change all the YES's to NO's. Copy II+ will overlook any checksum errors this way. Next, start from track $00, sector $00 and scan for these bytes:

AC 00 AC 00

In short this consists of periods and control-@'s. Write down the track of the occurrence. If you get a "string not found" error than this method won't work.

Next, go into the Bit Copier and set it to edit mode. For those of you who have Copy II+ press "/" when right before the copy process begins. Type in "0B" and <RETURN> then "02" and <RETURN>. By the way, only copy that track where the occurrence is found. Once the drive light stop spinning on the original drive type "f" (I think that is the command) and enter:

E7E7E7

If you see this pattern repeated many times than you have found the copy-protection. Starting from the first E7 (and including) count skip over six of them and press "C" when on the seventh. Now type the following:

AF F3 FC EE E7 FC EE E7 FC EE EE FC <RETURN>

Press "Q" to quit the editor mode and continuing with Copy II+ like normal.
DO NOT COPY ONTO YOUR ORIGINAL!!!!! That is a very unwise thing to do especially if the program doesn't use exactly this protection scheme.

Now, (if you like) got to you sector editor and read from that Track and Sector that you wrote down earlier. If your Custom Patch settings have not been changed

then read that spot. No error should occur unless a disk drive copied poorly.
Now, go back to the PATCH screen and change it to DOS 3.3 PATCHED. Escape back
to the sector editor (the place with all the numbers and characters) and write
the sector to you BACKUP disk.

Reboot, you are done. This works with many Broderbund, Epyx, and other programs
where Roland Gustaffson implemented his floppy drive routine.

If you have questions, comments, send them to mkelsey@eecs.wsu.edu

Background behind the unprotection scheme:

After picking apart the protection scheme I found the bytes that were being used
to protect the disk:

EE E7 FC EE E7 FC EE EE FC

These bytes can be shifted around and even changed to suit the purpose of the
author. Thus, this protection scheme is flexible and changes from program to
program. The Hex bytes above have almost become a standard.

When sychronizing to the disk the floppy drive uses sync FF's. These bytes have
a binary construction of so:

1111111100 1111111100 1111111100

 sync FF   sync FF   sync FF

The copy protection searches for the E7 bytes on the drive. Once it finds a few
it begins to read the copy protection. There is one limitation to the Disk II
Floppy drives. Zero bits (any more than two consecutively) are considered
invalid. Thus, raw bytes on the disk cannot start with zero bits. This
protection scheme is implemented because those zero bits, without special
hardware, cannot be read by conventional drives reliably. Specific programs
like Essential Data Duplicator 4.9 have the capability to control the write
process of the conventional floppy drive. This also works. But not everybody
has EDD 4.9, but most have Copy II+.

Any way, here is the raw bit structure of the Copy Protection

11100111 11100111 11100111 11100111 11100111 11100111
   E7   E7   E7   E7   E7   E7

Well by adding those zero bits, the bits that the drive can't read once
sychronized, the manufacturer is able to "fool" the floppy drive.

By using the AF F3 FC combination the floppy drive is forced to sychronize onto
the normal disk data and then reads like normal DOS or ProDOS.
The Copy Protection scheme jumps midway into a bit stream to catch the necessary
data. Thus, by adding one or two zero bits to the E7 byte patterns you can
obtain the EE E7 FC EE E7 FC EE EE FC data pattern.

For example:

Data read by a conventinal copier including the zero bits.
    / E7   **/ E7 \/ E7 */ E7 **/ E7 \/ E7 */ E7 */ E7 \
1110111001110011100111111001110111001110011100111111001110111001110111001111110 0
\ EE   / \ E7 / \ FC /\   EE / \ E7 / \ FC /\ EE / \ EE /\ FC /

   AF   F3   FC
10101111 11110011 11111100
    \   FF   / \ FF   /

Seeing as I didn't mention this earlier, the copy protection scheme jumps in midway into the seventh E7 byte. I placed 1010 to make the byte noticeable to other people and to be a filler space. At the fifth bit the drive will read the last four bits of the AF byte and the first four bits of the F3 and get an FF byte. The next to zero bits in F3 act to sychronize the drive. Next, the drive skips over the invalid zero bits and then takes the last two bits of the F3 byte and then gets the first six bits of the FC byte. This, once again, fills the eight bit data register and the last two zero bits force the drive to sychronize. I tried many methods of implementing this program. The first four zero bits 1010 in the AF byte were needed simply because the IIgs and //c have a different drive circuitry than an Apple IIe. With any more or any less bytes the copy protection scheme on the original would just fail. So, if you come across another method that generic copiers can duplicate (like this scheme) please let me know.

Understand that the "/ nn \" for the top, and the "\ nn /" for the bottom includes the bit that the slash is above or below, respectively. The top demonstrates the original's data stream with the zero bits below the "*". The bottom line illustrates how to obtain the special data from those extra undetect-able zero bits. Hope this hasn't been too confusing. If you need help or would like more information I can be E-mailed at:

mkelsey@eecs.wsu.edu

Note: if enough people ask for me write a program do this automatically then I will go ahead and write one up.

Back-up for Playroom (Broderbund)

Judy A. Neuhauser writes ...

> I am attempting to back-up our copies of Broderbund's program Playroom for Apple IIe. This is still in daily use in our Kindergarden classroom (Monarch Grove, Los Osos Ca.). We have contacted Broderbund but they no longer will sell disks of this program or support it in any way. Can a parameter file for Copy II+ version 8.3 be written which will duplicate our program disks? Of our original disks only a single copy is still working, Help!

A couple Computist articles dealt with Playroom-- one by Blackbird and one by Jack Moravetz. Both say that that you can deprotect the

program by first making a copy of Disk 1 using any whole-disk copier that will ignore errors. BB suggests Locksmith 6.0's Fast-copy. (Or, formatting a ProDOS diskette and copying the individual files from Disk 1 to the new diskette may work.)

The copy, of course, will not work. You need to change a few bytes on your new copy. Copy II+'s sector editor works fine for this.

If your copy is similar to the one mentioned you should find the stuff to change on Track 05, Sector 0A. If bytes $1F-$21 are 4C 47 64, cursor down to the first byte (using the I, J, K, M keys). When you're on the 4C, press H to get into Hex entry mode and type 18 [Return] 60 [Return] EA [Return]. Press ESC and Write the changed sector to the diskette.

The other diskettes should not be protected and can be copied with any copier.

Once you're sure the new copy works, you can make as many backups as you need using any quick diskette copier.

Rubywand

If you have Prince of Persia on 5.25", trade in for a 3.5". This is (or at least used to be) free. The 3.5" version is easy to deprotect:

Prince of Persia 3.5" deprotect:

Block: 7
Byte: $16
From: $90 (bcc)
To: $80 (bra)

Block: $A
Byte: $8C

From: $18 (clc)
To: $38 (sec)

This is to be used for archival purposes only.

It'll work from a 800K RAMdisk with this deprotect. Probably also a 800K HD partition.

Jay, krell@cornell.edu

I don't question the availability for a crack, but, I do question Br0derbund's usage of a quarter track in Prince of Persia. I, too, received the program as a gift (I still fell that it kills IBM P.O.P.(not 2) w/o a SoundBlaster) and I immediately set out to make my legal backup.

At least in version 1.0 (press CTRL-V during game play to check) there _IS NO_ quarter-track. You can tell quarter-tracks with true Apple Disk II drives. You can hear a slight tick instead of the chkkkkk....chkkkk....chkkkk normally heard when accessing track-to-track. I know this for a fact since this is how I detected the quarter track in AirHeart at 1B.25! In version 1.1 of P.O.P. there is the standard Br0derbund 'A5 96 BF' header and pitfully-easy-to-crack bitslip protection.

Avoid the bit-slip routine by searching track #$00 on the boot side for E7 E7 E7. I used Copy II+ (yes, a legal copy!) for any backing up. Using a nibble editor replace the seventh occurrence of the E7 pattern with AF F3 FC EE E7 FC EE E7 FC EE EE FC. This routine will cooperate with //c's and IIgs' unlike the PARM in Copy II+ Plus designed to backup CrossWord Magic (backups won't boot except on //e's because of machine specific timings).

Anyway .. .. .. that is the copy protection which exists on Prince of Persia. This same scheme occurs on Tetris from Spectrum Holobyte and Wings of Fury. Epyx also used the bit-slip timing in RoboCop, Ikari Warriors, California Games, World Games, Sub Battle Sim... etc...Br0derbund used it on Carmen Sandiego's (not W.I.T.W), Type!, and Animate. It was more fun to purchase the software just to crack it!

Let me know if stand corrected about Prince of Persia's copy-protection.

Michael Kelsey
mkelsey@eecs.wsu.edu

TD.CACHE: Cache mods by: Mr. Slick / Thanks: The Crasher

Test Drive II for the //gs is a very impressive game--except for its loading speed. The designers of the game decided to place the game's data in a lot of little files--which is the Achille's heel of ProDOS 16. Everytime ProDOS 16 goes to load a file, it re-reads the directory, and then goes and reads the file. This causes a lot of needless disk arm movements, and makes the game slow to load.

GS/OS solves this problem by caching the directory and so avoids the needless disk arm movements. Unfortunately, Test Drive II will not operate under GS/OS.

My fix speeds up the system using a caching technique.

INSTALLATION:

Copy the file TD.CACHE (in the unpacked ACU file) to a BACKUP, DEPROTECTED Test Drive II. It needs to go in the "/TD2MASTER/SYSTEM/SYSTEM.SETUP" subdirectory. Use the finder, copy II+, prosel, shrinkit, or such a program to copy the file. You don't need to copy the doc (TD.CACHE.DOCS) file.

You can backup Test Drive II using the deprotect available in the Apple II Games Forum, just copy the disk using Finder, Copy II+, or whatever and use a block editor and the deprotect to make it bootable.

Here's a chart of the time difference:

	Unmodified	Patched
Inital load	115 sec	51 sec
To go to Select Screen	28	7
To go to Race	65	20
To load Gas Station	40	14
To end Game	25	9
To go back to Select Screen	37	10

NEEDLESS DETAILS:

Why it works: TD.CACHE merely cache's the last 64K of data loaded from the 3.5" drive. It doesn't cache the directory per se, but with a buffer that size, the directory almost always stays in memory. In addition, it reads each track off of the 3.5" drive in one rotation, and all further reads for blocks on the track merely get data from the track buffer. Thus, it provides further speed enhancements.

I have a 64k cache installed. If you'd like a larger buffer (if you've got more than a meg of memory, make it as large as you can for even better
performance), do the following:

```
PREFIX /TD2MASTER/SYSTEM/SYSTEM.SETUP
BLOAD TD.CACHE,A$2000,T$B6
CALL -151

23C1: XX
23Ce: XX         These two addresses are the default ProDOS 8 buffer
                    lengths.  A value of 1 is 32K, 2 = 64K, 3=96K, etc.

23FF: XX         This byte is the default ProDOS 16 buffer length.
                    As above, 1 = 32K, 2=64K, etc.

BSAVE TD.CACHE,A$2000,L$1A00,T$B6
```

Mr. Slick

Question:
How do I fix Diskettes With Bad Blocks?
Answer:
The traditional wisdom in trying to salvage bad diskettes is: don't. There's a good reason not to if the cause of the bad diskette is bad block(s). A bad block is a defect on the disk media's surface. Do a disk verify in either Copy II+ v9.1 or the GSOS Finder and discover if there

are any bad blocks. The Finder tries 3(?) times to read a block and will report an error if it can't; I'm not sure how many tries Copy II+ or ADU makes. If any are found, discard the diskette. Bad blocks that show up on hard drives aren't as critical as on diskettes. R/W heads in a hard drive do not touch the surface of the HD's spinning platters, where they do with a diskette. A bad block on a HD can be "mapped out" safely during a low-level reformat or by using a separate utility with that function. The Operating System and file I/O software will then know not to access those blocks. If you try that mapping out process with a diskette that has bad blocks, the disk may last for a while, but eventually it will propagate more bad blocks as the R/W head(s) "drag" across the diskette's surface and extend the damage beyond the original bad blocks. The idea to try and salvage "bad" diskettes may be economically preferable, but the disks will eventually grow more bad blocks, destroying data in the process. I can personally verify the sad truth of this, having lost the only copy of several important files in the past. As I was a subscriber to several diskette magazines and software of the month clubs, the first thing I did was to verify each one and try to salvage as much as possible onto tested-good diskettes whenever a bad block was reported. About 5% of my subscriptions had bad blocks (the duplication process slows waaaay down if a verify is required, so most didn't do it). The salvage process usually did the trick, since most bad blocks were on unused sections or in non-critical files.

Deprotection information for:

Music Studio, Deluxe Paint //, PaintWorks Plus, Writer's Choice Elite, Top Draw, Print Shop GS, 816 Paint, Draw Plus, SoftSwitch, Odyssey

This information is provided only for use on legally purchasedly software for the express use of making archival backups. The University and the U-M Apple User's Group do not condone software piracy and provide this information for its responsible users.

Music Studio

Here's how to make a working backup copy of Music Studio that does not ask you to insert the master disk. In order to make an unprotected copy you need:

1) Any disk copy program that will ignore bad blocks on the disk. Copy II+ and Glen Bredon's volume copy program from Prosel will work fine.

2) A ProDOS block/sector editor. Block Warden (Prosel), Copy II+ sector editor, or the Bag of Tricks II Zap program will work. If you don't have access to one of these programs, a program that will do the job can be found in DL3 in the file PBE.EXE. The documentation for this program is in the file PBE.DOC.

To make the working copy:

1) Copy the original program disk to another 3.5" disk, telling the copy program to ignore the error on block 7.

2) Use the block editor to find byte $14 in block $44D (1101). You can also search the disk for the byte sequence: 0C 00 C9 01 00 F0, which will uniquely find the the proper byte.

3) Change the byte from F0 to 80.

4) Write the block back to the disk.

5) You now have a de-protected copy of Music Studio!

Several people have had trouble getting this program to work on a hard disk. Here are some tips to help out:

1) Make sure that you boot into ProDOS 16 from the hard disk.

2) Copy any files from the Music Studio /SYSTEM subdirectory that don't exist on the hard disk to the appropriate subdirectory on the hard disk. Make sure to delete the /SYSTEM subdirectory in the Music Studio subdirectory, as having two systems on the hard disk can cause problems.

3) If all else fails, the program should run from the root directory of the hard drive.

I would like to stress that these instructions are provided to allow archival backups only.

Deluxe Paint // Backup

Well, can you believe I found a common denominator in the Electronic Arts protection scheme that envolves Prodos 16 and Super Hi-Res Graphics..... Any way here is how to remove the protection on Deluxe Paint][

1. Copy the program key disk with Copy][Plus full disk copy (note you will have to format the target disk first).

2. Get out a sector editor like Prosel's Block Warden and get it up and running and working on your copy of Deluxe Paint][. Make it so that it will be reading and writing to that disk.

3. Read in Block $412

4. Enter the Edit mode and move the cursor to Byte $169

5. It should be over a byte that reads $A8.... Change this to $EA

6. Exit the Edit mode by pressing ESC and Write this block back out to the disk.

7. You now have a copy of Deluxe Paint][that you can back-up using the normal copy programs for the 3.5 drives.

Remember, this is only for making a working backup for yourself.

Paintworks Plus

Here's how to make a working backup copy of Paintworks Plus that does not ask you to insert the master disk. In order to make an unprotected backup you need:

1) Any disk copy program that will ignore bad blocks on the disk. Copy II+ and Glen Bredon's volume copy program from Prosel will work fine.

2) A ProDOS block/sector editor. Block Warden (Prosel), Copy II+ sector editor, or the Bag of Tricks II Zap program will work. If you don't have access to one of these programs, a program that will do the job can be found the PC5 library in the file UT.DISKWORKS. It is Shareware from Living Legends.

To make the working copy:

1) Copy the original program disk to another 3.5" disk, telling the copy program to ignore the error on block 7.

2) Use the block editor to find the sequence of bytes: C9 06 09 D0 01. This sequence is in block $291 (657). You can also have the program search the entire disk for these bytes.

3) Change the five bytes to EA's (NOP, or no-operation instructions).

4) Write the block back to the disk.

5) You now have a de-protected copy of Paintworks Plus!

Several people have had trouble getting this program to work on a hard disk. Here are some tips to help out:

1) Make sure that you boot into ProDOS 16 from the hard disk.

2) Copy any files from the Paintworks /SYSTEM subdirectory that don't exist on the hard disk to the appropriate subdirectory on the hard disk. Make sure to delete the /SYSTEM subdirectory in the Paintworks subdirectory, as having two systems in the hard disk can cause problems.

3) If all else fails, the program should run from the root directory of the hard drive.

I would like to stress that these instructions are provided to allow archival backups only.

Writer's Choice Elite

The following document describes the method for making a backup copy of Writer's Choice elite that does not require the use of the master (key) disk. To complete the archival backup procedure, you will need:

1) A disk copy program that is capable of ignoring bad blocks on a disk. Copy II+ and Glen Bredon's volume copy program from Prosel will do the trick.

2) A ProDOS block/sector editor. The Copy II+ sector editor, Prosel's Block Warden or Beagle Bros' Pro-Byter will work.

To make the de-protected backup:

1) Copy the original program disk to a blank 3.5" disk. If you use Copy II+, version 7.0 or higher and copy straight from the master disk to the new backup, it will automatically ignore the block 7 error. If you are using another copy program, you may have to manually force the error to be ignored.

2) Use the block editor to locate the sequence of Hex bytes: C9 07 00 D0 01. This sequence can be found in block $523 (1315), byte $73 (115). If the block/sector editor is capable of searching for hex bytes, you could have it search the entire disk for this sequence.

3) Change the five bytes to EA's (NOP, or No Operation instructions).

4) Write the block back to the disk.

5) You have now created a de-protected copy of Writer's Choice elite.

This procedure is intended to produce an archival backup copy ONLY of Writer's Choice elite. NOTE: If you like the fonts used in Writer's Choice elite, and you are using MultiScribe GS or DeluxePaint II, you can interchange the fonts between programs simply by copying each font file to the /SYSTEM/FONTS subdirectory of the program disk you want them on.

Top Draw

How to make an Archival copy of Top Draw. This information is furnished for the purposes of making a Backup of YOUR copy of the program.

First off make a copy of your program disk using any program that ignores block errors like Copy II+

Now search for HEX string 90 05 C9 11 00 F0 E6 AD 11 45 D0 EE

This was found in block $394 or for you Diskworks fans, #0916.

Change the EE to 00...

Seems to work great with the new desktop v3.1

PRINT SHOP GS

This information is offered for the purpose of helping those who PURCHASE a legitimate copy of Print Shop GS to make a backup for their own use, or to install Print Shop GS on a hard disk, so that it no longer requires a "master disk." It is a good (very good) program, reasonably priced, and worth consideration if you have a need for this type of program. Giving away copies of the program is not fair to the authors and publisher, who have produced a quality program, and deserve a fair return for their effort. If I ever get the idea that the information I am offering here is being misused, I will cease to upload any more help with de-protection of programs. Admittedly, I am as opposed to copy-protection as anyone I know, since it makes the programs so protected less valuable and useful; and, I do not appreciate publishers who use it, and believe it is time they start clearly labeling their boxes that their software is copy-protected. Yet there are reasons these publishers "protect" their disks, and I equally disapprove of unethical use of information, such as that I offer here, to "distribute" copyrighted programs.

That said, there is little difficulty in making a durable, normal

backup of Print Shop GS. Write-protect your original disk and copy it using ProSEL, Copy //+, or the System Utilities which came with your computer. There are no bad blocks on the disk, so the copying will go smoothly and quickly. It is not necessary to think in hexadecimal to ferret out protection code. It often begins with a PHA ($48) instruction, and will be found in either a loader type file or the main program file. The file MF on the Print Shop disk contains the protection. It begins with a PHA instruction, and terminates with the message "Please insert your master disk into the drive." I found it by searching the disk for the string "Please insert your master."

METHOD I:

Often a disk can be normalized by replacing the first instruction of the protection routine with a RTS ($60), and this is exactly what I did with Print Shop, and that proved sufficient to normalize it. I found the $48 in block $2F (dec=47), and used Block Warden to replace it with a $60. If it isn't there you can search your disk for the byte string $48 C9 05 00 F0 0F. Check to see that the routine ends with the "Insert" message, and zap out the $48 with a $60. This method makes one cosmetic sacrifice, as far as the program's operation is concerned. (Thanks to Walt Mossberg for pointing it out to me, as well as a typographical error in the hexadecimal notation of block #47, which is corrected in this upload.) The graphic image which normally is displayed with each menu choice when it is highlighted, never appears. You won't see the picture of the gs when "Setup" is highlighted, for instance, or the card with "Greeting Card." Instead, you get a blank gray rectangle.

METHOD II:

Leave the $48 at location $37 alone, or if you have changed it to $60 (using Method I), restore it to its original value. By doing this you are letting part of the protection routine run, because it initializes the graphics (and the way the mouse cursor interacts with them) which appear in the box on the main menu. The critical instruction, as far as the protection goes, is the JSR $674F, which appears at location $7F in block $2F. The exact code reads $20 4F 67. The code at $674F, which is the subroutine byte $7F calls when Print Shop is run, reads as follows:

```
674F: 20 5E 67   (JSR $675E)
6752: 20 5E 67   (JSR $675E)
6755: 20 5E 67   (JSR $675E)
6758: 20 5E 67   (JSR $675E)
675B: 4C 57 68   (JMP $6857)
```

The routine at $675E is a long one, which looks like a timing check to me. Whatever it is, it must be avoided if you do not wish to insert your master disk everytime you run Print Shop GS.

To accomplish this task the instruction at location $7F in block $2F must be changed from JSR $674F to JSR $6857. This will require that you zap two bytes. Change the byte at location $80 to $57. Then change the byte at location $81 to $68. DO NOT CHANGE THE BYTE AT $7F!!!!! There you have it, your cake and eating it too, so to speak. If you do not find these bytes at the above locations, search your disk for the hex string $20 61 42 20 4F 67. The last two bytes are the ones you must change.

Share your enthusiasm for Print Shop GS, not your program disk.

816 PAINT

Before I give you the specifics, I want to remind anyone who uses this information that it is offered soley for the purpose of helping those who PURCHASE a legitimate copy of 816 Paint to make a backup for their own use. It is the easiest to use of the full featured paint programs. And it is the quickest, as well, because the code is very compact. It is an excellent program, reasonably priced, and worth consideration if you have a need for this type program. Giving away copies of 816 Paint is not fair to the authors, who have produced a quality program, and deserve a fair return for their effort. If I ever get the idea that the information I am offering here is being misused, I will cease to upload any more help with de-protection of programs. Admittedly, I am as opposed to copy-protection as anyone I know, since it makes the programs so protected less valuable and useful; and, I do not appreciate publishers who use it, and believe it is time they start clearly labeling their boxes that their software is copy-protected. Yet there are reasons these publishers "protect" their disks, and I equally disapprove of unethical use of information, such as that I offer here, to "distribute" copyrighted programs.

Deprotecting 816 Paint:

The protection is in each of the two main program files, PAINT.320.SUPER and PAINT.640.SUPER. Baudville normally "disguises" the key elements of their protection code, changing MLI call commands from another place in the program, so that the call you see on the disk is not the call which is actually executed. In this case, what 816 Paint does is "attempt" to read a block(s) form the bad cylinder deliberately placed in the second position of the disk (Blocks #12 - #23). Currently, there are no copy programs available which will avoid formatting some of the blocks on a disk, so a disk like 816 Paint cannot be duplicated with the requisite bad blocks in place. If the read returns an error then the disk "passes muster" and the program runs. If it does not return an error, the program locks up. "Normal" code for such a check would start with the hex string $22 A8 00 E1 22 00.

Another routine apparently checks to see if the disk is larger than 800k, and if that is the case, ignores the signature check, so that the files can be run as is, from large capacity storage devices, such as hard drives, without inserting a "master disk." Baudville told me this when I ordered the disk, and it worked just as they represented it. I suppose, but have not actually tried, that a large enough RAM card would also enjoy this privilege. Baudville is to be commended, by the way, for allowing this. It represents a real step forward in copy protection schemes that are "somewhat" friendly to the purchaser. (The company is also very good about service.) If all you want to do is run 816 Paint from a HD, there is no reason to bother with what I am about to go into, unless you just want an extra backup for peace of mind. But, if you intend to run 816 from an 800k floppy, there is no substitute for doing so from a normalized backup.

So, looking for:

```
    22 A8 00 E1        JSL PD16MLI     ;entry point
```

```
    22 00               DC $0022       ;Read Block command bytes
    etc.
```

will not usually get the job done with Baudville. You must look for PD MLI calls that are odd, and displaced from purposeful and/or related code. On Award Maker Plus (PD8, not 16) the key call is an "Allocate Interrupt" call involving a bogus directory block!! They don't make it easy, but remember they could make it much tougher too. So don't pass copies of their program around, and maybe they won't esculate the protection wars.

STEP BY STEP:

Of course, don't use your block editor until you have made a copy of the disk, with a copy program such as proSEL or Copy //+, which ignores read errors. Then, to deprotect PAINT.320.SUPER read block $0276 (#630) into a block editor and find:

```
            05E: 22 A8 00 E1        JSL PD16MLI
            062: 06 00              (get info)
            064: CE AA 00 00
            068: AE 80 AF           LDX $AF80
            06B: 6B                 RTL
```

Change the byte at location $5E to $6B (= RTL), so that the routine never runs, no matter how the command code is altered by other parts of the program, as I can assure you they are. If this code is not in block $0276, search the disk for it, and change it if you find it. For PAINT.640.SUPER read block $02DE (#734) into a block editor and find:

```
            028: 22 A8 00 E1        JSL PD16MLI
            02C: 06 00              (get info)
            02E: 90 AC 00 00
            032: AE 4A B1           LDX $B14A
            035: 6B                 RTL
```

Change the byte at location $28 to $6B (= RTL), so that the routine never runs, again, no matter how the command code is altered. Remember, each of these files does its own check for the presence of bad blocks, so each must be de-protected.

Share your enthusiasm for 816 Paint, not their copyrighted code.

Draw Plus

Use a Volume copy program such as Prosel, Copy II plus, Diversi-Copy or whatever you like to make a copy of the Draw Plus disk. NEVER MAKE ANY CHANGES TO THE ORIGINAL DISK!!!!!!!! Ignore any bad block errors - these are the 'signature' blocks which will not be required by the back-up.

Using a BLOCK-editor (NOT a sector editor - this must be done using a 3.5" disk which is in the Pro-Dos format), scan for the following bytes (All values are in given in Hex NOT Decimal!):

1) $2B AD E8 0C C9 - I found this string at
   Block $516 location $2D

   Change the $2B to $00.

2) $23 AD 84 00 48 - I found this string at

Block $516 location $35

Change the $23 to $00.

3) $02 AB 60 E2 20 - I found this string at
Block $516 location $56

Change the $02 to $00.

4) $18 FB C2 30 0B - I found this string at
Block $516 location $A9

Change the $18 to 6B.

Now write Block $516 back to disk.

You now have an un-locked back-up of Draw Plus. This version may now be installed on a Ram-disk or a Hard drive.

Roger Wagner's SoftSwitch

The following is a method to free your GS from the copy protection scheme that RWP has seen fit to inflict on users of its new program SoftSwitch. This CP scheme is especially upsetting as it writes to a reserved area of your battery ram and comes from a company that has previously shunned CP.

Each time you cold or warm boot ProDOS 16, it runs certain files. When you install SoftSwitch a file called TOOL.SETUP.2 is installed in your SYSTEM/SYSTEM.SETUP subdirectory. This file, along with TOOL.SETUP is always run at boot time. An ID byte (or bytes) is also installed in the reserved area of your battery ram by the install program. After installation, each time you boot, the file TOOL.SETUP.2 looks for the ID byte. If it doesn't find it, SS will not be installed in the desk accessories menu. This makes the program only run on systems that have run the copy-protected install routine.

The key byte is $FB which is the very last byte of reserved battery ram. This byte is changed from a normal $FF to $FE.

In order to test out the theory that SS used battery ram as an ID check, I needed a routine that would read battery ram and place it in an area of memory that I could look at and modify. I also needed a routine to poke any changed bytes back into battery ram. The routines BRAMPEEK and BRAMPOKE [available on PC5 as UT.BRAMPEEK & UT.BRAMPOKE] give these capabilities:

1) BRAMPEEK

This routine will read the 256 bytes of battery ram from your system and put it in $2000 thru $20FF. I would suggest that you save these original 256 bytes to another file called BRAM.PARMS before you change anything (BSAVE BRAM.PARMS,A$2000,L$100) just in case you want to return everything to its original state.

2) BRAMPOKE

This routine pokes 252 bytes from $2000 forward plus a four byte checksum back into battery ram. I don't recommend that you change anything other than those bytes in reserved areas that SS modifies unless you want to risk inflicting a serious hangover on

your machine during the next cold boot.

Both of the above routines are intended to be executed by the AppleSoft "-" command after which you can enter the monitor by a CALL-151.

After running BRAMPEEK, look at the area from $2052 thru $207F and from $20A2 thru $20FB. These are the reserved areas of battery ram. If values in this range are anything but $FF, SS has altered your system. Change all the modified bytes back to what they should be and apply the following patch to the TOOL.SETUP.2 file:

1) Search for the following bytes: $A9 FB 00

2) Change $A9 to $80 and $FB to $51

This should let your system install SS each time you boot independent of any value of battery ram. If you find this patch useful, I hope you will take the time to tell RWP what you think about their SS CP scheme and urge them to remove CP from this and all future GS products. Thanks.

Odyssey: The Compleat Adventure

When I bought my Apple (early 1980), Odyssey and Zork I were the best games available. But you couldn't play Odyssey without Integer BASIC or 64K, so I never did until Softdisk put it on a disk in Applesoft ($10 - $5 coupon if you are a Softdisk contributor).

The game is still fun, but just too hard. I needed help. Like my drill sergeant said, "If you ain't cheating, you ain't trying." To cheat I had to defeat Softdisk's wimpy protection.

Anyway, if you have Demuffin Plus:

1. Initialize a disk.
2. Write a sector starting with 01 AD E8 C0 4C 59 FF or something to stop the drive & halt bootup to T0 S0.
3. Boot Odyssey.
4. Put in the noboot disk & press reset. It will load T0S0 to $800-8FF, but won't overwrite the protected DOS.
5. *2000<9000.BFFFM Move DOS out of way
6. Boot normal disk.
7.]BLOAD DEMUFFIN PLUS,A$803
8.]CALL-151
9. *9000<2000.4FFFM Move protected DOS back
10. *803G
11. Copy all files to blank disk with wildcard =.

To cheat, interrupt the program at any point with a ctrl-C, change variables (ie FM = men, BG & BI & Bsomething else control inventory)

Flight Simulator 2 v1.0

Hidden message:
Use a track/sector map utility, such as Copy II+ v5.0 or earlier, on the

disk to view a hidden message.