Open-Apple™

Releasing the power to everyone.

January 1986
Vol. 1, No. 12

ISSN 0885-4017
newstand price: $2.00
per page photocopy charge: $0.25

Does your mother love you?

"If your mother says she loves you," one of my journalism professors used to say, "*check it out!*" In the search for and explication of truth, journalists aren't supposed to use single-source stories, and they certainly aren't supposed to believe anything they read in the newspapers.

This professor retired from the journalism school at the University of Kansas last month. I wish this guy, John Bremner, would live and teach forever. His lesson, taught with fierce humor and gentle terrorism, is that "words mean." Writers and editors must understand words, and use them with order, logic, and common sense (as well as spell them correctly), or ideas are lost.

Though the Austrialian-born bear-like Bremner no longer teaches, his humor still professes in person and in his book *Words on Words, A Dictionary for Writers and Others Who Care About Words* ($7.95, Columbia University Press). The terror of sitting in his class next to someone who can't explain the difference between "careen," "career," and "carom," while knowing that your own browbeating comes next, however, is gone till the Resurrection.

If a student, for example, said that Applied Engineering's AppleWorks expansion software allowed 36 files on the desktop just because *inCider* said so, without checking it out further, Bremner would masticate him. I'm quite thankful that Applied Engineering and you subscribers are so much more forgiving—for it isn't so, even though I said it was last month. Applied Engineering has never modified, or promised to modify, that part of AppleWorks.

And if the same student also said that Applied Engineering's software allowed 16,300 records in a data base file, without actually seeing it done, Bremner would call him names he couldn't understand. Applied Engineering has promised 16,300 records, but hasn't yet delivered because of late bugs. Applied Engineering expects to have the feature ready by the first of the year. Free updates to purchasers of RamWorks II will be available for the asking.

Speaking of AppleWorks, Apple has released a new version. It's revision number 1.3. To get it, go to an Apple dealer and ask for an AppleWorks update disk-mailer. Put your original AppleWorks Program disk in it along with $20 and local sales tax and mail it. In return you'll get one 5-1/4 inch AppleWorks 1.3 start-up/program disk, one 3.5 inch start-up/program disk, and a 28-page addendum to the AppleWorks manual.

The enhanced version allows access to 3.5 inch drives in the same manner as floppies, allows you to format 3.5 inch disks from within AppleWorks, allows expansion of the desktop to 1,012K with an Apple memory expansion card, and allows the memory card or the extended 80-column card to be used as a RAMdisk. The other features many of us have come to expect from AppleWorks expansion software, such as larger data base and word processing files, are missing. None of the expansion software works with version 1.3 yet, but quick updates are expected.

Steve Wozniak is buying back into Apple. Press reports say he has purchased more than $5 million worth of stock recently. Wozniak, designer of the Apple II, sold all of his Apple holdings about a year ago (at a price more than half-again higher than the current level) because he didn't like the direction Apple was going. Now that Jobs is out, Wozniak likes Apple again—a very good omen for us Apple II people.

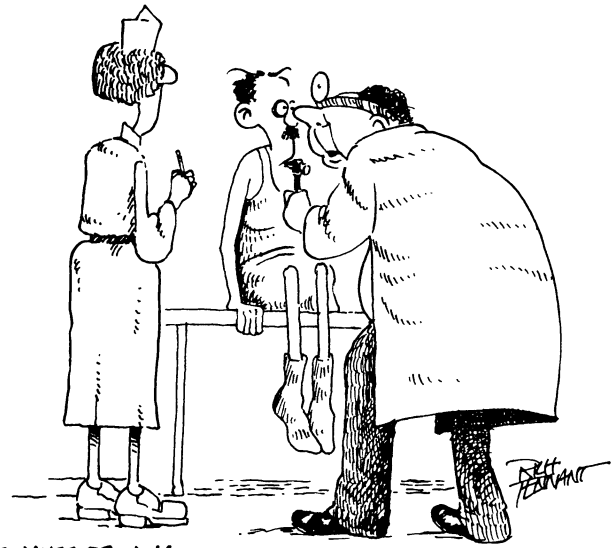
Wozniak is interviewed in the December/January issue of *II Computing*, a new magazine from the people who publish *Antic* for Atari machines. *II Computing* has attracted some very talented and II-wise columnists, including Margot Comstock, former editor of *Softalk*. Editorially, it is already a strong contender among the Apple-II-related magazines. You may be able to find it on newstands or you can get a 1-year, 6-issue subscription for $11.97 (PO Box 1922, Marion, Ohio 43306 614-383-3141)

The painters just arrived. Those of you who read Jerry Pournelle's monthly *Chaos Manor* column in ***Byte*** (*not* recommended reading for Apple II enthusiasts) know that he spent much of this year talking about the disruption of adding an office onto his house. I was jealous, so during the two weeks before Christmas we're having our house repainted and papered—while an ice storm rages and the kids are home from school and the ***Open-Apple*** deadline is pulling into what has suddenly become like Grand Central Station. Maybe I could fill page two with cartoons.

Robert Ericson's new book *AppleWorks: Tips and Techniques* ($18.95, Sybex Computer Books, 2344 Sixth St, Berkeley, CA 94710, 800-227-2346) is also disrupting my life. I got a copy this week and I've found it difficult to put down. The book is a greatly expanded version of Ericson's *Notes for AppleWorks*, which I mentioned in the August issue (page 61).

Bookstore computer racks are suddenly filling up with books on AppleWorks. Most of the books I've seen devote entirely too much space to information that's in the AppleWorks manuals. Ericson's book, on the other hand, supplements the manuals with literally hundreds of new tips, tricks, and intelligent observations. I recommend it highly.

Dealers will soon have a new version of *Apple Writer.* Take your original disks in for a free upgrade. (But keep a spare copy of the original at home, just in case.) The new version is called 2.1. According to Don Lancaster's column in the January *Computer Shopper* ($18/yr, PO Box F, Titusville, Fla. 32781), the changes are minimal. "None of the stuff that really needed fixed got done," Lancaster said.



"HE MUST BE A MACINTOSH USER. THERE'S A WRISTWATCH ICON ETCHED ON HIS RETINA."

The main change is that characters are now sent to the printer in high-value ASCII. This is how Applesoft and all DOS 3.3-based versions of Apple Writer do it. Version 2.0 of *Apple Writer* used low-value ASCII; this caused problems with several third-party interface cards. The new version still uses the ProDOS-standard low-value ASCII when printing to disk.

Some Apple DuoDisks can erase disks by accident. It happens during rebooting with open-apple-control-reset and with some copy protection schemes. Drives that have the problem have serial numbers beginning with 676 and ending with either 101 or 102. If you have such a drive, take it to your dealer for a free fix.

If the arrow keys on an add-on IIe numeric keypad don't seem to work, it's probably because the IIe has a revision C keyboard ROM, which turned up incompatible with keypads. Dealers are supposed to fix this without charge, as well.

There's also a bug in the new UniDisk 3.5. It doesn't return the WRITE PROTECTED error to ProDOS correctly. If a write-protected disk is in the drive and you try to write less than a block of data to a file that already exists, it's impossible to finish the write. It also becomes impossible to CLOSE the file. This causes problems later when programs try to reuse the buffer that was presumably freed up by the CLOSE. Apple intends to fix this bug by revising the firmware in the *drive's* ROM. Your dealer doesn't have this fix yet. Meanwhile, be very careful when you have a write-protected disk in a 3.5 inch drive.

A version 1.1 update to *SuperCalc 3a* has been released by the program's publisher, Computer Associates International. Call 800-223-9760 for more information. I've had serious problems with this program crashing into the Monitor—the last time it happened was right after I'd spent the better part of an hour typing in some data I wanted to graph. I just haven't felt the same about the program since.

Computer Associates' technical support people tell me I have a bad copy of the program—a hundred or so bad ones were sent out, they say, mostly for review and demonstration purposes (talk about putting your best foot forward). They insist that the production version of the program doesn't crash. I have a letter from a subscriber who says it does, even after he updated to version 1.1. I don't know what's going on—maybe he has one of the bad copies, too. But *SuperCalc 3a* hasn't earned a subdirectory on my hard disk.

Since I still wanted to graph that data, I switched to a review copy of *MouseCalc* I got from International Solutions (910 West Maude Ave, Sunnyvale, Calif 94086 408-773-0443). This sexy French import will produce nice graphs for you if you have the patience to battle the mouse to a draw. The program gave me a great appreciation for cursor keys.

MouseCalc uses double-high-resolution graphics for both graphs and the spreadsheet itself. This means it can display part of a spreadsheet and a graph on the same screen at the same time. The price you pay for this, however, is a scrolling speed that makes even a Macintosh look zippy. You just don't scroll this baby—you learn to use the mouse to slide the little "elevators" at the right and bottom margins to move around. Can you imagine how awkward and time consuming it is to use a mouse everytime you want to scroll a spreadsheet?

The program wouldn't boot on my mouse-equipped enhanced IIe with a color monitor. I didn't have time to figure out why. It didn't seem to mind running on my IIc, but I couldn't see the pretty colors. It's one of the few ProDOS-based programs I've run across that is copy-protected; a foolish decision on the publisher's part.

On the one hand, I feel that International Solutions should be encouraged for bringing graphically-advanced spreadsheet software to the Apple II family. On the other, I feel that a couple more programs like this one will send the Apple II mouse interface to a peaceful resting place.

If it happens, Apple's "evangelists" will choke in their pulpits while delivering the eulogies. In Apple's October newsletter for software developers, the "Apple Evangelist Manager," which I guess is sort of like Pope, went on about how he got depressed and stopped using his Apple II when the Macintosh came out because he "had a hard time dealing with its old-style interface." He's much better now, thank you, because of mouse-based Apple II software.

Then, in the November issue, the Apple II Consumer Evangelist talked about Apple's "long and determined effort to encourage development of Apple II software using that same (Macintosh-like) windowing environment." Later in the article he said, *"We want to see more mouse-based Apple II software,"* (italics his).

I think the mouse-interface is a very good one for certain specific kinds of software. Drawing software, software intended for occasional use, software intended for occasional users, and software intended for people who don't know how to type are the primary examples.

On the other hand, the mouse-based interface stinks for software intended for daily use by people who know how to type. Watch experienced Macintosh users—they avoid the mouse by using keyboard commands whenever they can. Being forced to use the mouse is disruptive; having it available slows program execution down. Apple's single-minded sermons supporting the mouse show a severe lack of perspective about how everyday users use computers. Most of their interface research was apparently done with new users.

A few years ago, when it appeared that the big market for personal computers was among the 90 per cent of the populace who had never used one before, the mouse was a good marketing strategy. Nowadays, however, when it's clear that people who already know how to use a computer buy more of them than new users do, pushing the mouse interface to the exclusion of everything else is bad for Apple and bad for software developers.

The best interface I can imagine is an AppleWorks-like magic-menu/command-key interface combined with an *Apple Writer*-like glossary and command language. (That description does seem a bit biased toward the software I use everyday, however.)

Reviewer's Corner

About 18 months ago I spent a day in agony because the power company switched my power off and back on four or five times. Each time I lost the data I was working on and had to reboot from scratch. If I had been saving something on a disk when one of these outages occurred (fortunately this didn't happen), I could easily have lost everything on the disk.

The experience was so upsetting I went out and bought an uninterruptable power supply. I'm happy to report that it has worked perfectly—the power company hasn't lost a single electron since I got it.

With the spread of hard disk drives in the Apple II community, uninterruptable power supplies are becoming more of a necessity than a luxury. There are many different kinds and sizes available at a wide range of prices, however, and selecting one can be difficult. But, as is typical in the Apple II world, the smallest and least expensive units are usually more than adequate.

The smallest backup power supplies I've seen provide 200 watts of power for 10 minutes or so—long enough to get you through most power outages completely and time enough to get your work saved and your system turned off if the power goes out for good.

To figure out how much power your system actually needs, make a list of the items you want to protect. Then look at the nameplate on each item to find out how many watts it uses. If the rating is in V.A., put it down as watts. If the rating is in amps, multiply it by 120 to obtain watts. Two-hundred watts is more than enough power for an Apple IIe with a monochrome monitor and a hard disk. Manufacturers do not recommend plugging printers into the backup unit, so don't even bother putting your printer on the list of items you want to protect.

A mail order outfit called Shepherd Marketing (P.O. Box 941339, Schaumburg, Ill. 60194, 800-AC-SPIKE, 312-894-5331) sells a 200 watt backup unit made by Tripp Lite for $249 plus shipping. This unit includes a built-in battery. Tripp Lite also makes, and Shepherd Marketing sells, cheaper models that come without a battery or battery cables.

PTI Industries (320 River St, Santa Cruz, Calif. 95060, 408-429-6881) makes a more expensive 200 watt unit, called the DataShield, that includes power line surge suppression and noise reduction. The Tripp Lite models lack this, but if you already have a Kensington System Saver or equivalent product protecting your Apple you don't need it anyhow.

PTI's DataShield also produces pseudo-sine-wave AC from the DC batteries, rather than the square-wave AC provided by the Tripp Lite products. This could be important for items that use AC power—most electrical motors, for example—but doesn't matter much if the item is just going to convert the power to DC anyhow, as your Apple II does. Square-wave is also sufficient for the Sider hard drive, according to technicians at First Class Peripherals.

Ask (or tell) Uncle DOS

Warranty war-front news

Your article on warranties appeared just as I was writing to Companion Software concerning their program *Facelift 2*. Their ad in the December *A+* said the program worked with the Epson FX, RX, LX, LQ-1500, and true compatibles. I ordered the program soley for its custom typefaces—its other features were unimportant to me. However, not until *after* I opened the package and read the documentation did I found out that the custom typefaces will not work on the RX (or on my RX-compatible Panasonic KX-1091). If I had known this before I wouldn't have ordered the program.

I have phoned the company twice. Each time I was told that the company has been having lots of complaints from Panasonic owners and that they would call me back. They haven't. As of today, I am stuck for $52.45. Surely one can successfully argue that a program's limitations must be made clear prior to purchase—even if the warranty says otherwise.

Arthur C Krohn
Palm Harbor, Fla.

I enjoyed the December issue's comments on software warranty disclaimers; they were right on, but not entirely accurate. The accurate part is that Apple's warranty disclaimer makes them sound unsure of the quality of their product and anxious to escape responsiblity. But to say it puts them in an unassailable legal position is wrong. Their disclaimer is not only wimpish, but also deceptive and ineffective.

When Apple (or anyone else) advertises a product as having particular features, they have made an express warranty. They cannot disclaim an express warranty by using words such as "as is" or by telling you that the risk is all yours. If their product (or anyone else's) does not perform as advertised, you have the right to get your money back.

Apple's disclaimer is probably effective in limiting their liability for consequential damages if AppleWorks reformats your hard disk, but I see no reason to accept the claim of their attorneys on that point.

William G. Tucker, Attorney at Law
Deep River, Conn.

When I took software publishers to task for their warranties in the November 1983 DOStalk, reader response was underwhelming. Nobody seemed very interested. My two-years-later blow-off about warranties in the December issue, however, seems to have hit a raw nerve. Software publishers, beware—your customers appear to me to be ready to hit the warpath over this issue.

Tucker's point is extremely interesting. If the authors of "as is" warranties know such warranties are unenforcable, then we are dealing with ***deception*** *here. Many unsatisfied software purchasers take no action because they assume the publisher's warranty is legally enforcable.*

You should **always** *attempt to get satisfaction when a program doesn't perform as advertised. The marketing and technical support people at many software companies will do whatever is necessary to keep customers happy—even if the same company's legal counsel says they won't—for marketing reasons. According to attorney Tucker, they should also be doing it because they are, in fact, legally obligated to—"as is" warranty or not.*

As for the companies that don't stand behind their products, let's hear your horror stories. There's no need for more than one ***Open-Apple*** *reader to get burned.*

IIc ID correction

I recently got the motherboard upgrade for my Apple IIc to correct the serial port timing problem. The technician told me that I also got the UniDisk ROM. You said that if the byte at $FBB3 is $00 it indicates the 32K ROM, while $FF indicates the original. I have a $06 at $FBB3. What is my problem? Do I have the new ROM or not?

If I do, how do I check to see if I now have the XON/XOFF protocol? And how do I access the new miniassembler?

Karen Lee
Westbrook, Maine

Uh,...my mistake. The correct ID byte is $FBBF (64447). That mistake is on page 89, change it now.

If you have the new ROM, you have the XON/XOFF protocol. You tell the IIc to start using it by issuing a PR#1, followed by "control-I X E". The X indicates XON/XOFF, the E means enable. A space character between X and E is optional on the IIc, but is required if you use this command on Apple's Super Serial Card for slotted IIs. To turn the protocol back off, use the same command but replace the E with a D, which means disable.

To get into the miniassembler, go to the Monitor with a CALL -151, then enter an exclamation point—the miniassembler prompt. Here's what happens:

```
]CALL -151

*!

!
```

Attempting to enter the miniassembler is an easy way to check for enhanced ROMs on both the IIe and IIc.

OPEN binary 3.3

I write educational software and am restricted to 48K machines, so DOS 3.3 (and ProntoDOS) will be my operating system for quite some time. Here's a trick to find out a file's size, and a lot more, without loading the file into memory or resorting to machine language calls.

When a file is OPENed, DOS 3.3 allocates space for and stores information about the file in the file buffers located between HIMEM and DOS. After a CLOSE command, the first byte of the filename stored in the buffer is cleared to indicate that the buffer is available for reuse, but the rest of the file's information, including a copy of the first track/sector list and the first data sector, is left intact. This data is free to use until the next file is accessed. If you have issued a MAXFILES 1 command, the data will even always be in the same place.

You can trick DOS into thinking all files are text files, so that you can OPEN them, by altering one byte in the routine at $A7C4. This routine checks the file type. If the first attempt to match a file's type fails, the lock bit is masked off and a second attempt is made. If you change the value of the mask to zero, every file will appear to be type $00 (text).

Beneath Apple DOS has a list of the file manager data items stored in the file buffer work area. Here's a simple program that can figure out how long in sectors a binary file is:

```
10 POKE 49254,0   : REM change "lock" mask
20 PRINT CHR$(4); "OPEN FILENAME"
30 PRINT CHR$(4); "CLOSE"
40 POKE 49254,127 : REM fix "lock" mask

50 FS=PEEK(46574) : REM get file size from buffer
```

I use this technique to calculate the highest possible loading address for binary data files before I load them and move HIMEM down.

Frank G. Andrews
Kalamazoo, Mich.

Even though I'm a connoisseur of cheap DOS tricks, I've never seen this one before. I think it has a lot of potential. Thanks for sending it in.

I now pronounce you

ProDOS supplies 128K Apple users with a RAM disk. The name, "/RAM", seems a bit sterile, however. It's like calling your dog, "DOG", or your spouse "SPOUSE". Here's a simple method for premanently changing "/RAM" into "/TOM".

The image of the RAM disk's name is stored at $2BD3-$2BD5 in the loaded ProDOS file (versions 1.1 and 1.1.1). The storage type and name length are both held at $2BD2. In versions 1.0.1 and 1.0.2 , two bytes must be subtracted from the above addresses. To make a permanent change, do this:

```
]CALL -151

*BLOAD PRODOS, TSYS, A$2000
*2BD2.2BD5
2BD2- F3 52 41 4D
*2BD3:54 4F 4D
*UNLOCK PRODOS
*BSAVE PRODOS, TSYS, A$2000
```

The three bytes at $2BD3 are the low-value ASCII values for "RAM." In this example, we change them to "TOM." The byte at $2BD2 has the storage type code for a volume directory in the high-order nibble, and the name length in the low-order nibble.

If your readers prefer a name other than "TOM", they can be as creative as they wish within the confines of three characters. If the new name contains one or two characters, change $2BD2 to $F1 or $F2. For the record, my first attempt at this resulted in a volume called "RUM."

Sandy Mossberg
Rye Brook, NY

You can also make temporary changes to the name of "/RAM", or permanent changes to the name of any other volume, with the Basic.system RENAME command. For example, RENAME /RAM,/TOM.

I found this out at an Apple user group meeting the day after I spent an hour trying to change my hard disk's name with a sector editor.

DoubleDOS disk tip

One nifty solution to the one drive DOS to ProDOS conversion is to use one disk which is half DOS 3.3

readable and half ProDOS readable. Making one of these beasts is not really that difficult.

What you'll need to start is one formatted ProDOS disk, one initialized DOS disk and a copy utility that will edit bytes and copy individual sectors (for DOS) or blocks (for ProDOS). To begin with, make sure that there are no files on either of your disks. This also means the HELLO file on the DOS disk must be deleted. Then copy track $11 from the DOS disk to the ProDOS disk (or blocks $088 to $08F if you are using a ProDOS disk editor). This moves the DOS catalog information to the ProDOS disk. Now file the DOS 3.3 disk away, it is no longer needed. The remaining disk becomes the doubleDOS disk.

The next trick is to set aside separate areas on the doubleDOS disk for each operating system. ProDOS stores its block usage map in track $00 sector $03 (block $006). To make ProDOS think that the second half of the disk is used (which indeed it will be—by DOS 3.3) change bytes $11 to $22 of that sector (or block) to $00. Now, to do the same thing to the DOS 3.3 part of the disk, change bytes $3C through $7B of track $11 sector $00 to $00. And that's it.

By copying a file (DOS 3.3 or ProDOS) to this disk the ProDOS convert program will transfer it to the other part of the disk when you only have one drive. This method is very handy for those of us (like myself) who even after reading your previous hints still can't figure out how to get CONVERT to work with one drive.

If another doubleDOS disk is needed these steps need not be followed again, simply use the ProDOS FILER or DOS's COPYA program to make another from the first.

Michael Woods
Granada Hills, Calif.

If you put PRODOS and CONVERT on the ProDOS section of the disk and change the name of CONVERT to CONVERT.SYSTEM, this disk will be bootable and will even run CONVERT on startup.

The other way to get CONVERT to work with one drive is to purposefully get it to create an error in the opposite direction from the one you want to go. For example, if you want to transfer from DOS 3.3 to ProDOS, start by telling it you want to go from ProDOS to DOS 3.3. Try to transfer a file, and you will get a NO ROOM ON VOLUME error. Go back to the main menu and reverse the direction of transfer to what you really wanted to do all along. Proceed normally, and CONVERT will suddenly start prompting you to insert the correct disk in drive 1. Apple has let CONVERT dangle without a fix for this bug (or several others) for over two years now (yet can't understand why many software developers are reluctant to use ProDOS).

800K disks, >128K *Apple Writer*

One of the things I like about ***Open-Apple*** is that I like to reread it. On looking at the October issue again, I was somewhat surprised to read, "Compare the 800K UniDisk 3.5 at $568 to a 10,000K Sider hard drive at $700 and tell me how Apple expects to sell these things." You mean, you don't know? I think it makes a lot of sense, what with the proliferation of expanded memory cards (whether the aux.slot type or Apple's own). As new programs and enlarged revisions of older programs come to market on 3.5 inch disks, we'll see mutual support for both areas of memory expansion. That can only make the Apple a more powerful machine—easier for the average person to make more powerful, too.

Now you're right that it's cheaper and faster to run a Sider instead. If all programs were copyable, you could simply transfer them to a hard drive. Eventually, the mass market will get to that megastorage point; if not with today's devices, then with some sort of laser system. But it seems like at this stage people feel more comfortable buying the program on a disk and sticking it in the drive. If that's the case, a high-memory removable disk drive is just the thing for souping up our "superannuated" computers.

Apple's new drive has a better chance to succeed in the current market and sometimes that's more important than state-of-the-art. Heck, Apple's already getting the software companies to cooperate. (Won't it be nice to have a spell-checker and thesaurus online, ready to pop-up as you write with good old *Apple Writer*?) And anyway, the price on UniDisks is sure to drop as the market for it grows.

Enough of my soapboxing. Instead, a question. Is there software for modifying *Apple Writer* (ProDOS) to handle larger files using a beyond-128K RAM card on a //e? If so, which cards would the software work with? I love *Apple Writer*, especially its glossary, WPL, and telecommunications functions.

David Hallerman
New York, NY

*I don't know of any software for expanding the **Apple Writer** "desktop." On 128K machines, **Apple Writer** can handle files more than 46,000 characters long. A whole issue of **Open-Apple** will fit in two **Apple Writer** files with room to spare. I don't think there's a pressing need among most users to expand beyond this, although I myself would occasionally use the extra space if it were available. It would come in handy for disassembling large machine language programs, among other things.*

Binary to EXECable hex

In your May issue (page 36), you showed us how to use a word processor to create an Applesoft program as a text file, which can then be EXECed into memory. You also showed us how to convert an *existing* Applesoft program into a text file, which can then be revised using a word processor.

Earlier, in February (page 13), you had shown us how to use a word processor to create a *machine language* program as a text file, which can then be EXECed into memory from the Monitor. Is there a neat way to convert an existing binary file into such a text file?

Paul Nix
Summit, N.J.

There are ways to put an EXECable machine language dump into a file, but none are as neat and simple as the other tricks you mention. Write an Applesoft program that opens a file and executes a WRITE command. Then, using the S.H. Lam technique from the February issue (pages 12-13), do something like C$="300.320": GOSUB 500. This will send a hex dump to the open file. (For a disassembly, try C$="300L".) Issue a CLOSE and end your program.

The big limitation of this trick is that Matthew Monitor's machine-language routines for dumping memory and Uncle DOS's machine language routines for writing on disks both use the same bytes on page zero for different things. As Matthew sends the hex dump to Uncle, Uncle stores the characters in a buffer until the buffer is full (256 characters for DOS 3.3, 512 characters for ProDOS). Then Uncle writes those characters on the disk. During the write, Uncle uses some bytes Matthew is also using. Matthew forgets what part of memory he was dumping and does strange and wonderful things.

Thus, if you use this trick, you must limit your dumps to short segments. Since a dump of 8 bytes takes 30 ASCII characters, don't try to dump more than about 135 hex bytes (about $88) into a ProDOS file, about 60 (about $40) into a DOS 3.3 file.

If you need to do very much of this, however, I suggest you read the next letter.

Dumper dug up

I have been unsuccessfully attempting to transfer a Monitor disassembly to a text file. I'd been able to feed the disassembled listing to the screen and to a printer, but invariably the disk file contents would be messed up after 100 bytes or so. I set the problem aside because I didn't have time to work on it.

Recently I was rummaging through my files and came across an article, "In the Dumps," that appeared in the October 1982 issue of *Call-A.P.P.L.E.* (pages 37-43); it is written by Val Golding with help from William Steinberg. It appears that the solution to my problem was in my files all the time. (Boy, I wish I could remember everything I've read!)

You'd told me in a note that zero-page conflicts were the cause of my file-writing problem. That indeed seems to be the case as Val reports running into the same difficulty. Steinberg provided the final clues to getting the program up and running. Val's short Applesoft program allows a hex/ASCII dump, a hex-only dump, or a disassembled listing to be sent to disk, screen, or printer.

The only hitch I've encountered occurred when I was toying with disassembling the higher reaches of the Monitor. I sought to send $FF30-$FFFF to disk and the program kept disassembling and writing to the file after it had passed $FFFF. There's probably a way around that problem, but this is a situation that more than likely won't be encountered during normal use of the program.

Paul Pagel
Enfield, Conn.

*I just found another widely-available program that will convert machine language programs into EXECable binary files. (It doesn't do disassemblies, however.) It's called AP/BIN TO TEXT and it's on the back of DOS 3.3 **ASCII Express** master disks. Unfortunately, it isn't on the ProDOS version.*

*Speaking of Val Golding, he's now editor of **On Three: The Magazine for Apple III Owners and Users**. Golding was editor of **Call-A.P.P.L.E.** during its golden years, so **On Three** should be good. A twelve month subscription is $40 (PO Box 3825, Ventura, Calif. 93006)*

Mopping up *Mouseprint*

Regarding the letter from Jim Willis in the December issue (page 95) about *Mousepaint* and Apple's continuing support (or non-support) for non-Apple products—a company called AHWARE (805 Luz Court, Danville, CA 94526) publishes two programs that modify *Mousepaint*.

The first, *Mouseprint* ($22.95), can be ordered for a specific printer and interface card combination. This allows you to print directly from *Mousepaint*.

The second, *Mousefont* ($34.95), allows you to change the *Mousepaint* fonts to any of twelve new ones, including icons. You also get an editor to create other fonts.

Both programs come without copy protection.

Tom Clune
Levittown, N.Y.

Boy, am I embarrassed. AHWARE had sent me data sheets on both products not too long ago. I must have been having mental hardware problems the day I answered Willis' letter; I forgot about the programs until you mentioned them. I have one slight ex post facto defense—the data sheet I have from AHWARE doesn't list Willis' PBI interface. But it has 19 others and could easily support the PBI by now.

80/40?

How can an Applesoft program tell if its host computer is in 80- or 40-column mode?

Michael Woods
Granada Hills, Calif.

I don't know of any sure-fire way to detect whether non-Apple 80-column cards are turned on. However, Apple's own 80-column hardware sets a soft switch at byte 49183 ($C01F) above 127 whenever the 80-column display is on. This switch is called READ 80COL. Try this subroutine:

```
200 SCREENWIDTH=40 : REM assume a 40-column screen
210 IF PEEK(64435)< >6 THEN 260 : REM II+/III test
220 IF PEEK(64448) = 0 THEN 240 : REM IIc test
230 IF PEEK(49175)>127 THEN 260 : REM IIe slot 3 test

240 IF PEEK(49183)<128 THEN RETURN : REM 40-col
250 SCREENWIDTH=80 : RETURN : REM 80-col
260 SCREENWIDTH=0 : RETURN : REM status unknown
```

Line 110 tests for an Apple II, II-Plus, or III using the standard ID bytes (see page 40). Line 120 tests for an Apple IIc, again using the standard test. If we get to line 130, we must have a IIe. This line tests for a non-Apple 80-column card in slot 3 by means of a rarely-used softswitch Apple calls the READ SLOTC3ROM switch. If we get to line 140, it means we have either a IIc or a IIe using Apple's own 80-column hardware. The test here is the crucial one. The routine returns with the variable SCREENWIDTH set to 40 or 80, or, if the status is impossible to determine, 0.

Pokes galore

Is it possible to access MouseText characters without activating the 80-column card?

Could you suggest a fairly simple technique to check, from Basic, if an 80-column card is installed? This is to prevent a program from hanging when a PR#3 is done without a card.

In the April issue (page 31), you have shown how to initialize a disk with 40 or more tracks, depending on the drive. Is there any way to automatically have the VTOC modified at disk initialization so that the extra tracks would be made available?

It would also be nice to make tracks 1 and 2 available when initializing instead of having to use a disk-zap program to achieve this.

Here's a tip for spreadsheet users. If you have a big spreadsheet you may have to wait awhile for recalculation. Instead of staring at your monitor waiting for the return of the cursor, press on the left arrow key a few times so that the cursor bumps against the edge of the spreadsheet when recalculation is finished. This will give you an audible reminder when you regain control and you can thus devote the intervening time to something more useful, such as reading ***Open-Apple***.

Paul F. McMullin
Campinas, S.P. Brazil

To access MouseText on a IIc or enhanced IIe without activating the 80-column card, simply do a POKE 49167,0 ($C00F). Any flashing characters already on the screen or sent to the screen will appear as MouseText. To turn them back to flashers POKE 49166,0 ($C00E). A program can tell if flashing or MouseText characters are currently displayed by peeking at byte 49182 ($C01E). A value of more than 127 in this byte indicates MouseText. See pages 26 and 27 in the April issue for much more about MouseText.

*You can check for the presence of most 80-column cards in an Apple II by looking at the card's identification bytes. These are at 49157+(SLOT*256), 49159+(SLOT*256), AND 49163+(SLOT*256). The values you find at these bytes should be 56, 24, and 1. (For any slot "s", those addresses in hex are $Cs05, $Cs07, and $Cs0B and the required values are $38, $18, and $01.) If these bytes have these values, it means there is a card in that slot and that the card follows the standard Apple card-identification protocol. (Not all do—look at slot 6, for example.) Now do this:*

```
CARD = INT( PEEK(49164 + (SLOT*256)) / 16 )
IF CARD = 1 THEN it's a printer
IF CARD = 2 THEN it's a joystick or mouse
IF CARD = 3 THEN it's a serial or parallel card
IF CARD = 4 THEN it's a modem
IF CARD = 5 THEN it's a sound or speech device
IF CARD = 6 THEN it's a clock
IF CARD = 7 THEN it's a disk or other storage device
IF CARD = 8 THEN it's an 80-column card
IF CARD = 9 THEN it's a network or bus interface
IF CARD =10 THEN it's none of the above
```

Once you've determined the firmware for an 80-column card is in slot 3, you can safely do a PR#3 on all machines except the IIe. On the IIe, firmware for an 80-column card is built into the machine—using it also requires a memory card in the auxiliary slot. The easy way to determine whether there is one is like this:

```
200 EIGHTY=1 : REM assume 80 columns
210 POKE 49153,0 : POKE 49237,0 : REM auxmem screen on
220 FOR I=0 TO 255 STEP 8
230 : POKE 1024,I
240 : IF PEEK(1024) < > I THEN EIGHTY=0
250 NEXT
260 POKE 49236,0 : POKE 49152,0 : REM auxmem screen off
```

If the routine returns with EIGHTY=0 there is no "80 column card" or other memory card in the auxiliary slot. This trick works by turning the 80-column memory on, placing some values in it, and seeing if they stick.

In your next question, I assume you want to make the POKE 46063,T (T=the number of tracks you want) mentioned on page 31 automatic, so that you no longer have to worry about how many tracks were on the disk used right before you initialize another one. Execute this series of pokes. They modify DOS so that it does the required poke itself everytime you INIT a disk. POKE 44697,169 : POKE 44698,T : POKE 44700,239. These pokes overwrite the useless volume number update DOS makes to the VTOC. (For more information, see the ***DOStalk Scrapbook****, pages 164-166).*

And here are three pokes that get DOS to initialize unbootable "data" disks with tracks 1 and 2 empty: POKE 42344,76 : POKE 44793,11: POKE 44723,4. For more information on this trick, see the ***DOStalk Scrapbook****, page 12.*

Grappler zap

Your November issue really helped clarify the interplay between software, printer, and interface. The need for standards in this area is compelling... but I think competition rather than reason will deliver the answer.

I bought a Grappler-Plus in mid-August of this year. It has a transparency feature controllable by both software and dip switch. This feature makes the card ignore all further control codes. You can turn it on with either a "control-I T" or "control-I Z". This is counter to a statement you made on page 86.

Bob Holdsworth
Wilbraham, Mass.

I'm glad to hear the Grappler people added that feature—it's not in my older card. I'd recommend you use the "control-I Z" version, since that is the command Apple's cards use for this feature.

Reset print buffer

I agree that AppleWorks should respond to reset, as you say in response to the letter "AppleWorks hates reset" in the December issue (page 95). But the problem the writer is having is not a problem with AppleWorks.

AppleWorks will quit sending a document to the printer (or buffer) if Escape is hit. Emptying the buffer itself isn't AppleWork's responsibility. Buffers that are built into printers can be emptied by turning the printer off. Stand-alone buffers usually have a switch for emptying the buffer. I know of one internal buffer that empties itself in response to a specific control-code.

To use it under AppleWorks, you press Escape to tell AppleWorks to stop generating output and then print a one-byte document that contains the buffer-flush control-code. You'd probably have to use the custom-printer code-redefinition tricks you've talked about many times to do this.

I don't know whether the Grappler-Plus bufferboard supports such a control code. I do know that the Grappler-Plus doesn't work with the print spooler in version 4.5 of Applied Engineering's AppleWorks Desktop Expander.

I agree with your comments about proposed MouseText versions of AppleWorks. I love the current AppleWorks windows. They are good. They shouldn't be screwed up with a Mouse. AppleWorks needs additional function, not interface.

Ken Kashmarek
Eldridge, Iowa

Rather than resorting to yet another custom-printer redefinition trick, you could instead keep an empty spreadsheet on the desktop that is set up to send the necessary buffer-flush code (using open-apple-O(ptions), SC). When you print this spreadsheet, AppleWorks will send out the special printer codes you have assigned, followed by a page worth of carriage returns or by a carriage return and a form feed (depending on how you have set the "accepts top-of-page commands" item in the printer specifications).

Program selectors

I read with interest your response to the letter asking for an automatic ProDOS program selector. You indicated that the best way to return to a main menu is to install your own custom dispatcher code within ProDOS. I agree, and I've given an example of such a program on pages 107-112 of my book *Apple ProDOS: Advanced Features for Programmers.*

By the way, you should point out that Applesoft programs and programmers can use the dispatcher

by entering the BYE command available in Basic.system 1.1.

Gary B. Little
Vancouver, B.C.

Another dispatcher is available directly from the author of the Merlin assembler, Glen Bredon (521 State Rd, Princeton, NJ 08540 609-924-5976—Note: the Post Office is mad at Glen because I said his address was 321 State Rd in the July issue, page 52.) This one is called /PROSEL and costs $20. I have a copy, but before I could get it up and running it got lost somewhere in my office. (There seems to be a black hole in here somewhere.) Glen's stuff is always good, however, so I'll recommend it freely.

Apple and Quark are selling a more expensive, mouse-based program selector called Catalyst 1.3. I haven't used it either. The ads make it look like yet another attempt to downgrade the Apple II into a Macintosh.

Apple's Pascal

I am a computer teacher who teaches Pascal programming and other things (AppleWorks, Basic, Logo). I have been programming in Pascal for four years, teaching it for three, and feel very confident with it. There are two problems with Apple Pascal 1.2 (128K) that I have not been able to cure or have corrected by Apple.

I am very concerned about the lack of support that I have been getting. Last May I wrote to Apple about the problems and have yet to receive a response. I have also called Vince Vezza, Apple's northeast regional representative, twice and have not had any response. A county-wide computer support person has also called Vince with similar results. This seeming lack of concern is very dismaying. I had hoped to at least get a response from someone.

I am writing to you hoping that you will be able to find out something or at least warn other users about the problems.

The first problem is in the editor. About two-thirds of the way through a large text file (30+ blocks) you will edit the line *above* the line the cursor is on. This is immediately obvious and usually does not do any serious damage. If the file is scrolled up and down a few times the problem seems to move to a different place but does not totally disappear. Once the problem has moved the corrupted line can be repaired. This is annoying but not fatal. A smaller text file always cures the problem.

The second is in my opinion much more serious. I have written a very large (66 block code file) custom data managment program. When the program runs the available memory decreases until it runs out and the program crashes. I believe the problem occurs in the managment of the data heap. If I do not let the operating system manage the data heap but take over management of the data heap myself with MARK/RELEASE, the problem does not occur. I monitor memory with MEMAVAIL and do not see any loss while I am in control. As soon as I turn control back to the operating system, however, *gradual* losses occur until STACK OVERFLOW. The only conclusion I have been able to reach is that the system is not returning all unused memory. I do not believe the problem is in my program since with MARK/RELEASE the program shows no loss.

I have tried to simulate the problem by filling up the memory with junk, but the system always works without a crash. I do admit the program is large, 3,500 lines of text, with very large data structures, but I don't think that is the problem. I do not use any dynamic memory allocation. I did at one time but rewrote the code when the problem surfaced. I do use recursion. To minimize the effect of hardware I used two different copies of original master Pascal disks and several different Apple IIes. All had the same problem.

I would appreciate any help you can give me. I would also like to know why Apple did not even respond. We use a lot of Apple equipment at the school where I teach and I hope the lack of support is not a forerunner of things to come. My local dealers (I have checked more than one) do not know much more about Pascal than how to boot the system, and some employees don't even know that.

Stanley Cauthers
Warwick, N.Y.

Like your dealers, I haven't ever gotton beyond Apple Pascal's prompt line, but—based on correspondence I've had with other Pascal users—I have some good advice for you nonetheless. ***Abandon Apple Pascal now.*** *Switch to* ***Turbo Pascal*** *under CP/M or* ***Kyan Pascal*** *under ProDOS.*

Apple has just released version 1.3 of its Pascal. Upgrading from version 1.2 will cost you $125 (buying it new costs $250). Compare this to Kyan and Turbo at $69.95. The support you get with Kyan and Turbo can't possibly be any worse. And you won't have to deal with Apple Pascal's uniquely obstructive operating system any more.

Apple Pascal has harmed the reputation of Pascal even more than FILER and CONVERT have harmed the reputation of ProDOS. I know that's a bold statement, but I stand by it. Apple makes many wonderful products, but some of its stuff reminds me of a poke in the eye.

No bold, wrong number

Just got the December issue of ***Open-Apple***, and it is really great. I'm thinking of dropping all of my other Apple subscriptions and buying three more of yours.

I have two questions for you. I am using AppleWorks with an IDS Microprism 480 printer. Before AppleWorks I used *PFS Write*, which allowed me to print in bold. Any ideas on how to do the same with AppleWorks? Second, page numbering messes up when I print more than 15 pages or begin printing with a page number higher than 20 or so on the second part of a document. I am professor and have a tendency to be windy and write long papers. Keep up the good work!

Gary R. Morrison
Cordova, Tenn.

According to Cardinal Point's book on printer commands (see the November issue, page 84), your printer doesn't have a boldface command. It does have a command that offsets a character slightly—if you do this and print the character again you can create a boldface look. This must be what ***PFS Write*** *does. AppleWorks doesn't support that kind of boldface.*

I've never used AppleWork's page numbering commands and don't know if they work or not. Help, someone.

More on printer control

In your November issue you say a sequence of PR#3, PR#1, and PR#0 has strange results. This is certainly true on an unenhanced IIe. The strange result is a screen display with every other column blank. However, the IIe enhancement kit solves this problem. You can add this to the list of reasons for getting the enhanced ROMs.

I have received some interesting mail since you published my letter on how to make the Okidata 92 work with AppleWorks (page 78). There seem to be two main problems.

First, the keys one must press to generate control-codes, especially the non-alphabetical ones such as control-\, are not understood. People will type control-\'s abbreviation, FS, and it doesn't work. Your ASCII control-code chart in the November issue (page 85) should help.

Second, the abilty to string several control codes together within an AppleWork's printer code isn't appreciated. For example, the answer to Jeff Russell's problem with the Olympia NP (page 87) is to place the Olympia's command for FINE ahead of the command for 10 characters-per-inch in AppleWork's 10 CPI printer code. The entire document will then be printed in FINE at 10 CPI.

Bruce W. Ristow
Rochester, N.Y.

Input-anything missing a BNE

I've been digging through your "Input Absolutely Anything" code from the September issue (page 68) and find it very instructive. There is one point that puzzles me, though. In the area from $349 to $350 you compare the location of a string's *current* storage area to the location of the *standard* string-storage area (FRETOP). Your goal is to force the string to be stored in the standard area if it is now stored within the program itself. First you compare the high bytes, and if string's current position is lower, you force it to be moved up. If the high bytes are equal, you check the low byte to make a decision. However, if the high byte indicates that the string's current position is *higher* than the beginning of the standard string storage-area (FRETOP), you still check the low byte. It seems to me this forces some strings to be placed at FRETOP when they could go back where they came from. Or am I missing something?

Stew Harris
Stamford, Conn.

You are right. Someday I'll learn to do a two-byte comparison with my eyes closed, but obviously I had trouble doing one with my eyes open last September. In the listing (page 68), there should be a BNE after the instruction at $34C. This BNE should branch to the TXA instruction a few bytes later. This is not a fatal error; it means the routine is just not nearly as efficient at preventing garbage collection as it was supposed to be.

Communicating with *Apple Writer*

This is a response to your "Call for *Apple Writer* calling help" in the October issue (page 78).

I have gotten *Apple Writer 2.0's* communications option to work, but not without some trial and error. I have a slot-resident modem, the SSM ModemCard from Transend Corp. Here's how I'm able to do it:

1. Using the Control-O ProDOS commands screen, I use option J, "set printer/modem interface." When prompted for slot, I enter 2, the slot my modem is in. The next prompt is for data format—I just press return to accept the default, but some people may need to change it.

2. Using the Control-P Print/Program command, I set the Print Destination to my modem with PD 2.

3. Using the Control-Q screen of additional functions, I select option I, "Connect keyboard to printer/

modem." At this point, I activate the modem normally, but skip PR#2 and IN#2—that's what Control-Q (I) does. In the case of the ModemCard, I type Control-A, Control-Q and am prompted by the modem's own software to enter the phone number. Hitting return then dials the phone.

4. Once the connection is established, there are two choices. For bulletin boards, things proceed pretty much as they do inside any other communications software. But uploading a file—which is my principal use for AppleWriter's communications facility—is a little different, and this is why we set the Print Destination to 2. I load the document to be uploaded into the main AppleWriter screen memory; then Control-P plus NP sends the contents of memory out through the modem.

5. Disconnecting again is done as though you were working directly with the modem in BASIC.

I'm not sure how any of this generalizes to other slot-resident modems, but it works for me. I found that Apple's generally wonderful documentation fell down on this portion of AppleWriter, but that their generally wonderful software was up to par once I figured out how to use it.

Owen R. Youngman
Chicago, Ill.

To log online to CompuServe with *Apple Writer*, I set up a glossary entry for use with my Hayes Smartmodem 1200. After the initial key code (I use the mnemonic "C" for CompuServe), it reads: control-O J2 1200,8,N,1)] control-P PD2) control-Q I ESCAPE E AT DT XXX-XXXX]

The control-O takes me to the ProDOS Commands menu. "J" is the menu seletion for "Set Printer/Modem Interface." "2" is the slot my modem interface is in. "1200,8,N,1" refers to baud rate, data bits, parity, stop bits (the commas are necessary here). "]" means two carriage returns.

The control-P takes me to the Print menu. "PD2" is the print destination—my modem in slot 2. Again, "]" is the glossary's return character.

The control-Q takes me to the Additional Functions Menu. "I" is the menu selection for "Connect Keyboard to Printer/Modem." The ESCAPE E sequence keeps my keystrokes from appearing twice on the screen (echoing). "AT" wakes up my Hayes modem, and "D" tells it dial, and "T" tells it to use touchtones. "XXX-XXXX" is the local CompuServe phone number. Finally, one last "]" carriage return is needed to enter the phone number. When I receive the "Connect" message, I need to hand-enter "Control-C," my ID number, and my password. I haven't yet figured out a way to do that with the glossary. Does anyone know how?

The spaces in the glossary line above are not needed; I've included them only for clarity's sake. If you're creating this glossary line after pressing "control-G ?", then you only have to press control-O, control-P, control-Q, and ESC for those codes. But if you're making this up as a text file, each of those characters needs to be embedded in the file by bracketing it with control-Vs.

You could type all these keys by hand to use your modem, but that's the beauty of the *Apple Writer* glossary.

David Hallerman
New York, NY

I am sorry to report that the above techniques do not work with my Hayes Micromodem. When I get to the Control-A, Control-Q part, the control characters appear on my screen; the Micromodem doesn't respond to them. The same thing happens when I press Control-P.

My Micromodem, which is the pre-IIe version, also messes up the 80-column screen display.

*I don't know what's going on among software developers, but there is a sudden proliferation of new software that boldly proclaims on the outside that it includes a "communication module" but that bashfully hides in the documentation that the module works only with modems hooked to a serial card (or with a slot-resident modem that acts like a serial card). (Two packages I otherwise like, **Pinpoint** and **Swyftcard**, are among these.) Since most of the modems in use in the Apple II world are slot-resident Hayes modems and the equivalent, I think the user community should be screaming fiercely about this. At the same time, I have to recommend that anyone planning to purchase a new modem give some weight to this software problem when deciding which modem to buy.*

*While playing once again with **Apple Writer**'s communication option, I did notice that pressing escape followed by any other control character changes the communications-command-key — something else the documentation neglects to mention. This could be a handy feature if you needed to send an escape character over the phone.*

*As I read the **Apple Writer** manual, the first step the two writers above mention may not be required. I don't think it tells **Apple Writer** where the modem is—I think it's for telling the the serial card how to send data to the modem (or printer). As I understand it, your response to the data format question generates a setup string for the serial interface card. This string, incidentally, is invisibly saved inside Print Value files. If I'm right about this, I should have mentioned it in November's section on solving **Apple Writer** printer problems (pages 87 and 88), but didn't.*

Scrapbook whys and wherefores

In the ***DOStalk Scrapbook*** you give programs for finding the file manager parameter list and the IOB. The book *What's Where in the Apple*, by William Luebbert, on the other hand, says the file manager parameter list is at $B5BB-B5D0, and the IOB at $B7E8-B7F8. These are, in fact, the values I obtained when I ran your programs. Why do we need the programs? What additional purpose do the programs serve?

I have a very short program from *Call -A.P.P.L.E.*, by Bill Parker, that reads and writes disk sectors. Why is yours so much longer?

Page 150 of your book the fifth line says JSR $3E. Shouldn't that be JSR $3E3?

Vedula N. Murty
Middletown, Penn.

DOS 3.3 was designed to run on an Apple II with as little as 16K of RAM memory. Luebbert's addresses are the correct ones for DOS when it's at the standard location for machines with 48K or more. Since such machines are all that's left these days, the programs for finding the file manager parameter list and the IOB appear unnecessary. Some people with 64K or larger machines, however, like to use special programs to move DOS 3.3 up into the extra 16K of RAM. This leaves almost the entire lower 48K available to Applesoft—and leaves Luebbert's addresses dangling in mid-air.

*The programs in the DOStalk Scrapbook were written to cover contigencies such as these. They work no matter where DOS 3.3 is located. Programs like the one you mention from **Call -A.P.P.L.E.** work fine when DOS is at its standard 48K location—which may be all the time for some people—but fail otherwise.*

Likewise, the additional commands in the DOStalk Scrapbook's Sector Reader cover other contigencies. The text explains what each line does and why it's necessary (at least it was supposed too; if we skipped something let us know).

You're right about the typo on page 150. The book was combed for typos by three different people but it just seems impossible to get all the bugs out of something this big. Other readers have pointed out these errors:

On page 130, the comment in the last line of the program should say $D9, not $09.

On page 157, in the second paragraph, change the reference about line 130 to line 155 and the reference about line 145 to 165. Change other references about line 145 on this page to line 285. The program the text refers to was changed slightly after the text was written. Since the text refers to line numbers the program doesn't have, many readers suspect there are lines missing from the program listing. Fortunately, this isn't true—the program is complete, I just forgot to revise the text after revising the program.

*In the compendium of DOStalk pokes and patches on page 245, the last entry on the page should start $9D56-9D61. In the book, the second D in this entry is a zero. At the top of the same page, in the section on the Page-3 Vector Table, byte $3EF (1007) should be shown as "unused." A new entry should be added for $3F0 (1008), which is the **address** of the break handler. (I used to think this was a **jump** to the break handler, but that's not how the Monitor uses this location.)*

On page 246, in the comments on the routine at $A744, the reference to $9000 should be to $9600.

The book is available now in bookstores all over Kansas City and no doubt elsewhere, too, for $14.95. The publisher is pushing a hardcover version by mail order for $6 more. Please note that many bookstore employees don't seem to know what category the book falls in. My wife found it with the books on UNIX in one store. Feel free to move it to the Apple section in your own favorite bookstore and put it on the shelf with the cover, rather than the spine, out. Thank you.

Track 0 crashes

The business department at the high school where I teach chemistry got 24 128K enhanced IIes this fall. We also have about 60 older IIes in the school. Five classes a day have been working with the ProDOS version of *Apple Writer* on the 24 new machines—very little else has been done on these computers so far.

Shortly after the school year began we started having problems with the new machines. Though everything worked properly most of time, several times a day students would try to load a file and get a VOLUME NOT FOUND message. By the time I was called in, they had about a dozen of these disks for me to look at.

Each was ruined the same way. Using a DOS 3.3 sector editor, I found block 2, the directory, and block 6, the volume bit map, were gone on these disks—including the address and data marks. It looked as if all the sectors on track 0 had been wiped out. The files that the students had saved were, for the most part, still there and recoverable by reformatting track 0 and rebuilding the directory.

The problem has occurred at all times of the day, on most of the computers, with different students, and with different copies and versions of *Apple Writer*. Some of the drives and controller cards were replaced with older Disk II drives and some failures occurred

on those as well. We metered the electricity for 48 hours and found nothing unusual.

At one point I was called to look at one computer that seemed to be consistently malfunctioning at that moment. *Apple Writer* was running. I destroyed several newly formatted disks, once by trying to save a file, once with a Catalog, and once with a List Volumes On Line. I then rebooted *Apple Writer* and it worked fine.

We have called Apple and they have not been very helpful. Everything being used in our school is an Apple product—it seems like the problems should have occurred elsewhere. I can't think of anything that makes our situation unique.

J. Ernest Cooper
Lathrup Village, Mich.

It appears ProDOS is **destroying** *track 0 when it should be* **reading** *it. You're right, this isn't typical.*

So, it means something is destroying ProDOS, which in turn destroys track 0. My first guess would be that the copies of Apple Writer you are using were generated from a single bad master disk. You say you've used different copies and different versions, however, so that's probably not the source of the problem.

The other possibility is that the hardware itself on these machines is mysteriously changing bytes in the memory area ProDOS lives in. Since the computers were all purchased at once it could be possible that you have machines from a bad batch. Try putting a few older IIes in the classroom—if the error also occurs on one of them it would indicate a software problem. If no errors show up on the older IIes, then the new machines probably have some subtle hardware problem.

I'm surprised Apple isn't sleeping at your school trying to figure this one out. Not many companies will spurn a $100,000 account like yours—to say nothing of the 120 potential customers in those classes who keep having their disks erased. I think your school district should be more assertive with Apple. If you can't get help from your local people start bugging Cupertino.

Open-Apple

is written, edited, published, and

Open-Apple
P.O. Box 7651
Overland Park, Kans. 66207 U.S.A.

ISSN 0885-4017. Published monthly since January 1985. Worldwide prices (in U.S. dollars; airmail delivery included at no additional charge): $24 for 1 year; $44 for 2 years; $60 for 3 years. All back issues are currently available for $2 each; seven or more from any single volume $14 (postpaid). Index mailed with the February issue. ***Open-Apple*** is available on disk for speech synthesizer users from Speech Enterprises, P.O. Box 7986, Houston, Texas 77270 (713-461-1666).

Source Mail: TCF238 CompuServe: 70120,202 Tele.: off hook

ProDOS TYPE command

In the June 1984 **DOStalk** in *Softalk* (page 157) you published a public domain TYPE command for ProDOS. Is it possible to buy a disk with this program on it? I'm very interested, but have no desire to do all the typing.

Bruce Kahn
Stony Brook, N.Y.

The program is in most Apple User Group software libraries on International Apple Core disk #44. If you don't have quick access to a group, you might want to look at **Apple Software for Pennies** *by Bertram Gader and Manuel V. Nodar ($9.95 + 1.00 handling charge from Warner Books, PO Box 690, New York, NY 10019). The book is a directory to a significant part of the public domain software available in Apple User Group libraries. It gives the addresses of several groups that will take mail orders from anyone.*

More memory puzzles

Will Apple's new memory card allow Apple Writer IIe or AppleWorks to run on a II-Plus? Will software written for a II-Plus/IIe MIDI music interface run on a IIc with a MIDI interface?

Quest I. Oners
Name Adr, Lost

No, Apple's memory card works in a II-Plus, but it doesn't turn it into a IIe. See the August issue, page 57, for a reference to a program that will get AppleWorks going on a II-Plus. I don't know anything about the MIDI interface. I suggest you call the publisher of the program or the manufacturer of the IIc interface and ask.

Apple Writer null patch

Here is a patch for the DOS 3.3 version of Apple Writer IIe that allows you to insert control-@ (null, hex 00) in Apple Writer documents:

```
CALL-151

BLOAD OBJ.APWRT][E
1CE1:EA EA EA EA
3EBA:EA EA EA EA
UNLOCK OBJ.APWRT][E
BSAVE OBJ.APWRT][E,A$1900,L$2F58
LOCK OBJ.APWRT][E

BLOAD OBJ.APWRT][F
1D81:EA EA EA EA
4033:EA EA EA EA
UNLOCK OBJ.APWRT][F
BSAVE OBJ.APWRT][F,A$1900,L$3001
LOCK OBJ.APWRT][F
```

The patch was sent to me by the folks at the Apple Regional Technical Support Center here on the west coast after a couple of weeks of talking to my dealer, Orange Micro, and Star Micronics.

Mike Reaves
El Cajon, Calif.

Let's add that patches should be made only on ***back-up*** *copies of your original disk.* ***Apple Writer IIe*** *is lightly copy protected, so this can pose a problem.* ***Do not let the copy protection tempt you into modifying your original disk.*** *Situations like this one are what programs such as* ***Copy II Plus*** *are written for.*

Amiga watch

How do you think the Apple II will fair against Commodore's Amiga?

Wardell De Loach
Camden, N.J.

If Steve Jobs and the Macintosh couldn't dent Apple II sales, how can Commodore do it? Jobs even had the power to stop proposed Apple II enhancments so the II wouldn't embarrass his machine. Commodore hardly has the power to convince its bankers to let it stay in business. The Amiga is nice, but don't look for it to lower Apple II sales. It may hurt the Macintosh, however. The only machine around at this time that will steal sales from the Apple II is the new Franklin.

International characters

Having seen only your issues for November and December I was struck immediately by two things: the technical expertise of the publication, and the enjoyable writing style. How refreshing to find someone who is "computer literate" and also literate in the more traditional sense. ***Open-Apple*** is free from the misspellings, grammatical lapses, typos and other blemishes that so often mar one's pleasure when reading materials by computer mavens (including—or should I say especially—hardware and software manuals).

Here are two rather specific questions relating to both printer problems and AppleWorks.

I often write in Portuguese and Spanish. I set up AppleWorks' custom printer codes so that control-B produces a backspace instead of boldface, thus causing the accent mark to print over the correct letter. However, the next line is then offset one space beyond the left-hand margin; this offset continues until the next Return (end of the paragraph). Why?

It would be of enormous help to me if I could use the AppleWorks Replace function to globally replace every #, say, with Spanish or Portuguese words that include accent marks. But AppleWorks' Replace doesn't allow control characters (the backspace, control-B, mentioned above). Is there a patch for AppleWorks to get around this? Apple Tech support was not at all supportive when I approached them with this question. Can you help?

Clifford E. Landers
Montclair, N.J.

Once you execute the AppleWorks control-B(oldface) command, the assigned control character is sent at the beginning of each line until the end of the paragraph. This is what is causing your first problem. The solution is to make AppleWorks think you've turned boldface back off by pressing control-B a second time after you type the accent mark. Tell AppleWorks your printer's code for "boldface end" is nothing at all.

AppleWorks' inability to Replace (or even Find) control-codes is a serious problem. I use the same function in ***Apple Writer*** *to find or change a sequence of characters that includes the Return control-code frequently; I've always considered the inabilty to do this in AppleWorks a handicap of using that word processor. One possible solution is to put the words that include the accent marks on the clipboard, then F(ind) the # marks and C(opy) the accented words from the clipboard one position at a time. This works, but it involves more keystrokes than a good Replace function would.*