Open-Apple

February 1988
Vol. 4, No. 1

ISSN 0885-4017
newstand price: $2.00
photocopy charge per page: $0.15

Releasing the power to everyone.

Mainframe genie at your service

Those of you who have been subscribing to ***Open-Apple*** for awhile probably remember the cartoon in our June 1986 issue, in which a fearsome looking IIc genie is saying to a shaking human at the keyboard, "Yes, Master?" It's always been one of my favorites.

Christmas morning (I know it's strange to talk about Christmas in the *February* issue, but the January issue was already printed by Christmas morning, so I couldn't possibly tell you this story in *it*) the kids woke Kathy and me up and we crept downstairs together. It got kind of steamy about half way down; by the time we got to the Christmas tree it was obvious that someone besides Santa had been there. I went to the fireplace and reached for my stocking. As I touched it, there was a loud humming sound, like a modem pushing 2400 baud. Suddenly smoke was everywhere. Carolers were at our door singing out passwords. A booming voice shook the house with, "Tom, let's form a strategic alliance." I was sore afraid. It was the mainframe genie.

From a pocket deep within his flowing robes he pulled out a clipping from the November 30th *InfoWorld* and asked me to read it. I started to say that Steve Gibson's column was the only thing in *InfoWorld* worth reading, but thought better of it. The clipping said that Apple was planning an on-line communication network for Apple users. The network would be managed by a company called Quantum Computer Systems, which has a similar product for Commodore users. Quantum would operate Apple's service, the clipping said, which would be sold under Apple's logo. The new network would offer technical support from Apple itself. Subscribers would use Quantum's own communications software to access the service.

"Sounds great," I said. "All the II-family genies will be able to get together on line and release the power to everyone."

But as we talked, I could tell that this mainframe genie wasn't happy. The smoke started to settle and his robes slowly became translucent. Through them I could see a three-piece suit. Then it dawned on me. This wasn't just any mainframe genie, this was GEnie, the online network operated by General Electric. GEnie had lost its Apple II manager to Quantum and wanted ***Open-Apple*** to take his place. "No," I said, "I don't have time."

"We'll pay you royalties," the GEnie said. "They're not enough," I responded and negotiations were open. We wrestled. Mainframe genies are strong, but not overpowering. I fired up AppleWorks and attacked him with memos and spreadsheets. "I know how to deal with these suits," I thought to myself.

The GEnie was about to raise its one-time sign-up fee for new users to $29.95 from $18. I demanded no sign-up fees at all for paid ***Open-Apple*** subscribers. "I'll do it," the GEnie said, "but no free time and no free manuals" (the $29.95 includes a manual and two hours of free time).

I already liked the basic rate for connect time on GEnie; $5 per hour for non-prime access (evenings after 6, mornings before 8, weekends and holidays) at either 300 or 1200 baud; $12.50 per hour for 2400 baud. More importantly, there are no minimum monthly charges. None of the major national networks has lower rates; most are higher. GEnie can be accessed with a local telephone call from almost 500 cities in the U.S. and four in Canada. "What about our international subscribers?" I asked. "I'm a multinational company," GEnie responded, "expanding around the world."

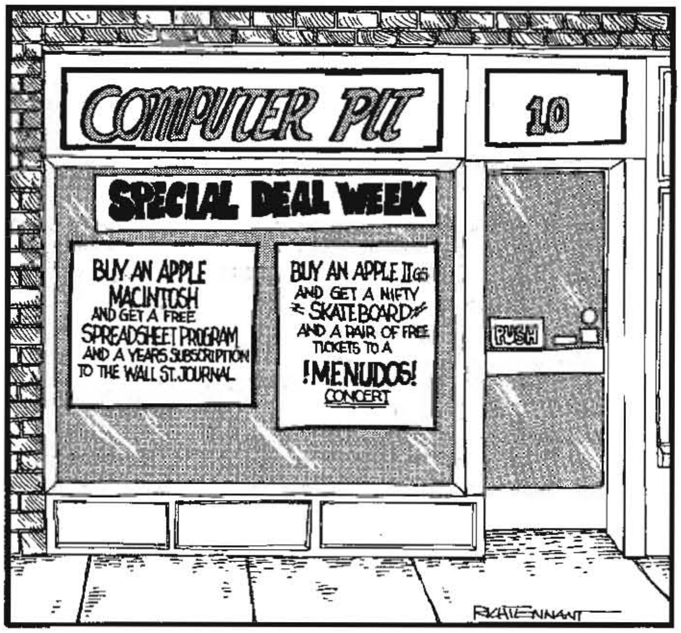
Self-actualization was starting to set in. Like the other national networks, GEnie offers news, shopping, airline schedules and fares, financial information, and electronic mail. And, like the other national networks, the most popular part of GEnie is its bulletin boards and libraries devoted to computers. On GEnie these areas are called Roundtables. There are two active Apple II Roundtables on GEnie–a general one and a second one for programmers and developers.

Each Roundtable has its own bulletin board, conference rooms, and library. The bulletin boards are organized into "categories;" the categories into "topics." Unlike other systems, a user's message isn't deleted after a few days, but stays on until the area manager deletes it. Thus, the bulletin board consists of a large number of "threads," each on a specific topic, with tips, questions, and answers going back, in some cases, two years or more. As negotiations progressed, I started to think about editing these threads so that the older information in them was all solid, while the newer information toward the bottom of the thread would contain today's questions and more tentative answers. I could imagine hundreds of Apple II users working together to release the power, each in his or her own areas of interest. Then I realized I was imagining an issue of ***Open-Apple*** that had come alive.

The conference rooms in each Roundtable can be used for online meetings. A good example of the power of conferences is the Binary II file format for making Apple II communications easier (see "Sending files by carrier bunny," April 1987, page 3.17). Gary Little used the conferencing feature of one of the national networks to discuss Binary II with other Apple II users interested in communications protocols. These discussions led to improvements in his proposed format and to its rapid acceptance in the Apple II community once it was finalized.

The libraries in each Roundtable hold public domain software and text articles, as well as many shareware programs that can be examined in the comfort of one's own home and purchased by sending the author the necessary fees. Pictures and sounds also make up a portion of the Apple II libraries.

As we wrestled, I was never able to persuade the GEnie into paying royalties at a level high enough that I could hire a professional to manage the day-to-day operations of the Apple II Roundtables–even if the professional I hired got every penny. But financial considerations have never been very high on the list of priorities around here, and the potential of a GEnie/***Open-Apple*** alliance for releasing the power of the Apple II eventually overcame me and I signed on the dotted line.


COMPUTER PIT
10
SPECIAL DEAL WEEK
BUY AN APPLE MACINTOSH AND GET A FREE SPREADSHEET PROGRAM AND A YEARS SUBSCRIPTION TO THE WALL ST. JOURNAL
BUY AN APPLE IIGS AND GET A NIFTY SKATEBOARD AND A PAIR OF FREE TICKETS TO A ¡MENUDOS! CONCERT
PUSH
RICHTENNANT

It will be a few weeks yet before this alliance is settled and organized. I'm still learning how to use GEnie's system. I've been to an introductory conference in the Apple II area where the GEnie regulars (many of them already ***Open-Apple*** subscribers) threw electronic pies at me. The procedure for ***Open-Apple*** subscribers getting free accounts isn't settled yet, but it will involve calling a GEnie 800-number with your modem and providing your name and credit card number for billing purposes. Stay tuned.

Starting with this issue, ***Open-Apple*** will carry bits of information about the software available in GEnie's Apple II libraries, news about upcoming conferences, and tips and tricks gleaned from its bulletin boards.

If you're a GEnie user already, you can start sending mail to me (UNCLE-DOS) and to Dennis (OPEN-APPLE) right now. See you online.

Miscellanea

Claris changed from egg to chick in mid-January. I've given Claris a mighty hard time the last few months, but we have to live with Claris now, so I'm declaring a unilateral cease-fire.

Claris started shipping a "slightly revised" version of AppleWorks in mid-January. The only revision I can detect from talking to them over the phone is that they changed the disk label. They're calling it AppleWorks version 2.0. This in itself is pretty special; you might remember that when Claris was announced, Apple said Claris wouldn't be allowed to use Apple trademarks.

To its credit, much of the Claris's early efforts appear to be going into customer service. It says it intends to provide customer support, which will be quite a change from how Apple itself handled AppleWorks. The Claris version of AppleWorks will retail for $249. Upgrades from earlier versions of AppleWorks are $75. For more information on the upgrade, call 800-544-8554 (8:30-5:30 M-F Pacific time) and ask for an AppleWorks upgrade packet. They'll mail it to you; use it to mail your check and original disk (as proof of purchase) back to them.

If you already have AppleWorks 2.0, save your money until we find out if there are any real changes here. If you have an earlier version of AppleWorks, this is your chance to get the latest. According to the Claris representative I talked to on the phone, the upgrade can be made from any prior version of AppleWorks. Availability of the upgrade expires September 30, 1988.

According to the Wall Street Journal (January 11, 1988, page 5) the upgrade program "is not merely to win customer goodwill. When Apple itself handled the marketing and support chores for the programs, it didn't bother to keep returned warranty cards from software customers, much less lists of their names. Claris's discounted upgrade 'will help us identify those customers,' said Mr. Zeisler (John Zeisler, vice president of marketing), who contends that successful software companies cultivate repeat customers. 'We really wish we knew their names already.'"

Maybe Zeisler would like all of us who aren't going to buy the AppleWorks upgrade to send him our names (along with our wish lists of what we'd like to see in future AppleWorks versions). If you do write to him, remind him that about 80 per cent of his revenue is coming from Apple II programs and about 90 per cent of his development dollars are going into Macintosh programs. Then tell him what you think about that. His address is:

```
John Zeisler
Claris Corp.
440 Clyde Ave.
Mountain View, CA 94043
```

If you'd rather just call and leave a message, the corporate phone number is 415-960-1500.

Time stopped for ProDOS 1.1.1 and earlier incarnations of ProDOS on January 1 of this year. Clay Ruth warned us this was going to happen way back in October 1985 ("Time short for ProDOS," page 1.79). Sorry I forgot to re-mention it a couple of months ago.

ProDOS 1.1.1 and earlier versions were designed to work with a clock card (the Thunderclock) that didn't keep track of what year it was. To compensate for this, the clock routines in ProDOS 1.1.1 figure out what year it is by matching the day-of-the-week with the date. It's all very complicated, but it works....for seven years. Then it doesn't.

Since ProDOS 1.1.1 won't read the clock in the IIgs anyhow, if you have a IIgs the point is moot. If you have any other Apple without a clock in it–same result. If you do have a clock in any other Apple II, however, you'll notice that all the files you've changed since the first of the year now have a last modification date of 1982. If you try to use AppleWorks, you'll find that it won't even accept 1982 when you try to get past the startup date-prompt. (One wonders whether Claris updated the version of ProDOS on its new AppleWorks disks–Apple's last AppleWorks 2.0 disks still had ProDOS 1.1.1 on them.)

The best solution is to get ProDOS 1.4 from your dealer or from a friend who has a IIgs. On the IIgs system disk, ProDOS 1.4 is in a file in the SYSTEM folder and is called P8. You need to rename it PRODOS and copy it into the root directory of all the disks you boot with. While you're at it, you might as well get the latest ProDOS system utilities, too. They're in a folder called SYS.UTILS on the IIgs system disk and will work on any 64K Apple that can display 80-column text.

ProDOS 1.4 solves the track 0 crash bug we discussed at length in November 1986 (see "ProDOS bug found in Australia," page 2.73) and is the version we now recommend using. We do not recommend either ProDOS 1.2 or ProDOS 1.3–both have bugs relating to the use of two 5.25 drives, among other things. Incidentally, the track 0 bug described in November 1986 was *always* confined to 5.25 inch disks. That bug has nothing to do with any 3.5 disk, hard disk, or RAMdisk problem you may experience.

If you don't have ready access to ProDOS 1.4, another alternative is to patch ProDOS 1.1.1 as described in Ruth's letter back in 1985. A public domain program by Glen Bredon that installs a similar patch is in file #3047 in the GEnie Apple II library. These patches only work till 1992, however.

We've finally had time to start digesting the Beagle Bros *Timeout* applications around here. They're all good. *Ultramacros* is incredible. But *Quickspell* is a work of true genius. What makes *Quickspell* so great is its interface. You prepare your document within AppleWorks. You press open-apple escape to bring up the *Timeout* menu and scroll to and select *Quickspell*. It immediately checks through an internal dictionary of most-used words, your personal custom dictionary, and an 80,000-word Random House Concise dictionary. Then it lists on the screen all the words in your document that it couldn't find in any of the dictionaries. You can scroll through this list and select all the words you want to ignore. Then you can scroll through it and select all the words you want to add to your custom dictionary. You can also choose to replace misspelled words in the list. The cursor point-and-select interface is at its finest in *Quickspell*.

(The Macintosh interface supports a similar system, but you have to use the mouse button to both scroll the list and to select individual items. In the Apple II interface, the up and down arrows make the cursor scroll and the right and left arrows select or deselect items. You can also select a single item with Return. On the Apple II, it is very easy to control the speed at which the list scrolls and to select individual items. On the Macintosh, the list scrolls too rapidly to stop exactly where you want and using the mouse button for two different functions causes lots of extra hand movement. Don't expect to read about this in Apple's *Human Interface Guidelines*, however.)

At any time, *Quickspell* also lets you look at your misspelled words in context, in which case you see the whole AppleWorks screen with a small *Quickspell* menu at the bottom. From the in-context display, you can once again choose to replace a word, ignore it, or add it to the custom dictionary. But from this screen you can also request a suggested spelling. *Quickspell* will then display a list of words spelled similarly to the one you have misspelled. Again, you can scroll through this list and pick out the spelling you want with a simple Return.

Quickspell can be set up so that it checks only a single word or screen rather than a whole document. It also checks for double words.

The two big problems I've always had with spelling checkers is the amount of time they typically take to use and the large number of correctly spelled "misspelled" works they find. *Quickspell's* three-dictionary speed, the ease with which you can deal with multiple words at one time, and the ease with which you can add words to the custom dictionary solve all these problems. I love this program. I now find myself spell-checking not only ***Open-Apple***, but even the simplest, shortest messages. (*Timeout Quickspell*, $69.95, Beagle Bros, 6215 Ferris Square, San Diego, CA 92121 619-452-5500.)

Updates for other programs in the *Timeout* series all require that you send in your original disks as proof of purchase, a $2.50 handling charge, and...for *AutoWorks*, *MacroWorks*, or *SuperMacroworks* to *UltraMacros*–$20; for *SideSpread*, *SuperFonts*, or the both of them to *FontWorks*–$10, $20, or $30, respectively.

Turning Point Software now has an upgrade for *Time Is Money* that allows it to work on an Apple IIgs. Back in August ("Cheap accounting," page 3.51) I promised a review of accounting programs, but I haven't gotten very far on that project. We have tried Manzanita's *BusinessWorks*, but stopped looking at it when we discovered it throws away all your transactions every month and just keeps account totals. This is how professional accountants do it, I guess, but a seat-of-the-pants operation like ***Open-Apple*** needs a little more flexibility than that. I also bought *DAC Easy Accounting*, but it was so slow once we got it set up that nobody around here would use it. So we're still using *Time Is Money* ourselves. For more on this program, see the last paragraph of my response to "The Sider and Software" in September 1985, page 70. (Upgrade $15, program $100, Turning Point Software, 230 Western Ave, Boston, MA 02134 617-782-4877).

One of the least understood selections in the Apple IIgs control panel is the very first one–the "display type," for which there are two choices, color and monochrome. The user manual's advice, which is essentially to use the "color" selection if you have a color monitor and the "monochrome" selection if you have a monochrome monitor, is simplistic.

The manual doesn't mention it, but this selection does totally different things depending on whether your monitor is connected to the 15-pin RGB video output or the round "RCA-type" composite output. The composite output is what other Apple IIs use. On the IIgs, this output does exactly what it does on other IIs when the IIgs "display type" is set to "color."

When "display type" is set to "monochrome," on the other hand, all color signals coming from the IIgs composite output are changed into gray-scale signals. This slightly modifies the dot patterns that represent colors on a monochrome screen from splotchy to regular, in all graphic modes. Internally, what the control panel actually does is change bit 7 of the IIgs Monochrome/Color Register, which is at $C021 (0=color, 1=gray scale). Remember, however, that this register affects only *composite* monitors.

If you have a composite monitor (either color or monochrome), our advice is to leave the control panel's "display type" set to "color;" this will give you exactly the same screen image that you'd get with your monitor if it was hooked to any other II. The gray-scale option does have the potential to clean up your screen image slightly, so you may like to try it from time to time. But software designers who put any thought at all into how color images look on monochrome monitors will be working from the standard II-family display; since most IIgs owners buy RGB monitors, hardly anyone has a system that actually supports the IIgs gray-scale capability.

With the more popular RGB monitor, the color/monochrome selection *does* have an effect, but you have to be in exactly the right place to see it. That place is double-high resolution graphics. Apple created standards for RGB double-high-resolution graphics for the Apple II family when it released the Apple Color 64K Extended Memory card for the IIe. These standards were discussed here in ***Open-Apple*** way back in May 1985, page 36, and July 1985, page 54. On a IIe or IIc equipped with an RBG monitor, there are three double-high RGB modes. These are color (140 x 192 in 16 colors), monochrome (560 x 192 in black and white), and mixed (a combination of the first two). Keep in mind, however, that these three modes are available on a IIe or IIc only when it's equipped with an RGB monitor; this, too, is rare. Without RGB, IIe/IIc double-high-res appears as 560 x 192 black-and-white on a monochrome monitor and as 140 x 192 16-color on a color monitor.

Unfortunately, Apple didn't follow its own RGB standards on the IIgs. An RGB-equipped IIgs supports color and monochrome double-high modes, but not the IIe/IIc mixed mode. In addition, the method for switching between color and monochrome modes is completely new on the IIgs. On the IIe and IIc, RGB double-high-res mode-selection involves tickling several softswitches in certain specific sequences, as demonstrated here in 1985.

On the IIgs, on the other hand, RGB double-high-res mode selection is determined by bit 5 of the Video Control Register, at $C029 (0=color, 1=*monochrome;* if you change this bit from inside a program, use a read/modify/write procedure so that you don't accidentally change the other bits in the register)

As mentioned, changing the Monochrome/Color Register at $C021 affects only the composite output, but changing the Video Control Register at $C029 has an effect on both the composite and RGB outputs. This is a little strange because there never has been and still isn't any such thing as a double-high-res monochrome mode for *composite* screens (double-high-mono is an RGB-only mode). But if you are looking at a color composite monitor with the Monochrome/Color Register set to color and start flipping the Video Control Register between color and monochrome, you will see a color shift on your display. It means the best strategy for programmers is to keep the two switches synchronized, that is, either both set to color or both set to gray-scale/monochrome. The easiest way to do this is to enter the IIgs control panel and toggle the "display type."

The *Desktop* program from early IIgs system disks changes this bit to turn on monochrome double-high-res. Notice that if you enter and exit the control panel while using *Desktop,* the screen display turns to multi-colored mush if display type is set to color. You can play with this to convince yourself that the color/monochrome setting really does do something on an RGB-equipped IIgs.

There are many doubters, however, because the setting works *only* in double-high-res. Many new RGB-IIgs users are discovering problems with II-family software that uses standard (rather than double) high-resolution graphics. The problem is that many graphics programs were written by and for II/IIe/IIc users who had composite monochrome monitors. On color monitors, the nice blacks and whites that these programs are supposed to display turn into color smears. If you use the control panel to switch to monochrome, nothing happens. You can fix all this, however, by tricking the IIgs into displaying double-high-resolution monochrome mode and then looking at just half the picture.

Setting up an Apple II to display double-high-resolution graphics requires that you hit the softswitches at $C050 (graphics), $C057 (high-res), $C05E (double-res), and $C00D (80 col). To look at just half the picture, go back to 40 columns (hit $C00C). Now, if you have the IIgs control panel set to monochrome, you will see a standard resolution graphic on a IIgs RGB monitor without the color blurs. It's quite simple to modify existing 40-column Applesoft programs to use this "new mode." Simply add a POKE 49246,0 (double-res on) to the beginning of the program. Now, as long as the control panel is set to monochrome and your program stays in 40 columns, you will see a monochrome single-high-res picture. To return to a standard-res color display, either change the control panel setting or POKE 49247,0 (double-res off). This trick comes from Apple's *Apple IIgs Technical Note #29,* so we can expect it to work with future members of the Apple II family, too.

The "first final" version of *Apple IIgs Programmers Workshop C* is now available from the Apple Programmers and Developers Association for $75. Updates to version 1.0 from beta versions are available for $17.75. The program requires the Apple Programmers Workshop version 1.0, 1.25 megabytes of memory, and two 3.5 drives or the equivalent.

CMS has a new ROM for its SCSI hard drives that corrects some problems when using ProDOS 16 programs such as the *Finder.* The new ROM is free to any CMS owner by calling 714-549-9111, ext 216.

Somebody took a search and replace sword to the January 1988 edition of the *E.A.C. Express,* the newsletter of the Erie Apple Crunchers user group in Erie, Pennsylvania. Every place the word "Apple" should appear there are instead five underline characters. The editor explains:

"A major concern among user groups during the past year was the implementation of licensing agreements for our computer's operating systems and the use of a certain company's logos and trademarks. The position taken by this company on the licensing of the operating systems is quite fair for all parties involved. However, the permission to use the trademarks and logos comes with quite a few strings attached. Basically, we are supposed to put the little 'TM' mark after the first appearance of a product for which this computer company holds a trademark. We are also supposed to use the little 'r' in a circle and so on and so forth...We will try to comply with the trademark and logo restrictions as much as possible in future issues. It's too bad that _____ did not put a 'TM,' or 'c' in a circle, or an 'r' in a circle in their printers!"

Magnets in speakers placed too close to the IIgs analog RGB color monitor can wash out or warp the colors on the screen. Just moving the speakers farther away from the monitor doesn't help immediately–you also have to turn the monitor off for several minutes before it will correct itself. This tip comes from the January edition of the Washington Apple(TM) Pi newsletter.

The *Desktop Publishing Newsletter* is a new publication that will feature techniques for producing effective publications on the Apple II for schools and businesses. It will be published by Sage Productions, the same company that publishes the *AppleWorks Journal.* The premier issue is expected in February 1988, later issues will follow every other month. A one-year, six-issue subscription is $20 (5677 Oberlin Drive, San Diego, CA 92121 619-455-7513).

Apple has been working hard to make big sales to the federal government. The problem is that most federal contracts require MS-DOS compatibility. The slots in the Macintosh SE and Mac II were supposed to solve that problem. But Apple's "first major victory" in the federal marketplace, according to Charles Berger, Apple's vice-president of market development, was the sale of 4,000 Apple IIgs systems together with 4,000 of Applied Engineering's PC Transporter cards. The computers will be used in 270 schools for dependents of U.S. military personnel. The company that selected the Apple IIgs for the government said that it had received proposals from several computer suppliers and that Apple was chosen because its bid represented the "lowest life-cycle cost." I hadn't thought of that before, but the Apple II does have a much longer life-cycle than your standard MS-DOS machine. The IBM-PC has come and has been discontinued, yet we still have the Apple II.

As we begin volume 4 of *Open-Apple,* we're making a few adjustments to our operation (in addition to the GEnie alliance) that I thought you might like to know about. One problem we've had is that you, our subscribers, insist on buying stuff from us. Now that, in itself, isn't so bad. But we've never been set up to provide you the kind of service you expect from a high-tech computer-related company. You expect 24-hour order turn-around. Our system here, however, is modeled after what's typical in the magazine industry–four week turn-around if there are no snags. You've complained about it, I've complained about it, and it has come down to either getting out of the book and back-issue business or getting into it with all six feet.

Our current level of sales doesn't justify all six feet. However, if my own experience at trying to buy Apple II-related books is any indication of what's going on in the world, the kingdom is badly in need of a mail-order retailer who carries a large selection of Apple II books. We've decided to expand and improve our mail order business and to try to fill that niche. We're working on an order entry, inventory, and shipping system now and expect to be in full operation by mid-summer. As part of the change, we're bringing our subscription list back in-house (see May 1986, page 2.25) and we'll resume processing it with Apple IIs. Unfortunately, all this has required that I give up being a cottage industrialist. On December 1, **Open-Apple** began moving out of peoples' houses and into 2,000 square feet of office/warehouse space in a top secret location somewhere east of the Rockies.

An office needs to have people working in it to amount to much. Two of our work-at-home consultants agreed to become office-based employees; one wouldn't leave the cottage. Still with us are Dennis Doms and Sally Dwyer. The other Sally, Sally Tally, our imaginary Circulation Manager, may or may not be moving here from the fulfillment service in Syracuse N.Y. that we're about to stop using; she hasn't decided yet. We have added the locally-renowned Apple II and Mazda wizard Tom Vanderpool to our technical-side staff. And our business-side staff now includes Steve Kelly, an MBA who comes to us with very rare and nearly unbelievable business credentials–he's been a Kansas farmer for more than ten years and has a net profit to show for it.

Open-Apple has never had to pay rent before, postage rates go up this spring for the second time since we started, and each direct mail campaign we do gets a slightly lower response because our best prospects are already subscribers. I've had to face the reality that Open-Apple's subscription price must go up–an ugly prospect. This will happen March 15, 1988. Our new prices will be $28 for 1 year, $54 for 2 years, and $78 for three years; free airmail worldwide, as usual. Renewals at the old prices ($24, $44, and $60) will be accepted through April 1.

Facing and accepting the financial limitations of Open-Apple has been difficult, but accepting the philosophical limitations of my take-no-prisoners position on copy protection (August 1987, page 3.50) has been gut wrenching. The problem with my position is that it restricts the amount of software available for the Apple II (and thus the value of the Apple II) in two ways.

First, obviously, those of us who simply refuse to buy or use any copy-protected software blind ourselves to a major share of what's going on in the kingdom. Secondly, and far more importantly, Apple II software developers deserve to be compensated for their work. The ethic that it's all right to "make a copy for my friend/colleague/students" pervades the kingdom. That ethic is wrong. If the pervasive ethic was that asking someone to copy one of their programs for you was the same as asking them to shoplift a memory card, we would be in a better position to demand no copy protection. But that's not the pervasive ethic. Have you ever noticed how much bigger and wealthier Apple II hardware companies are than Apple II software companies? Now you know why.

I am still against copy-protection. I will always buy or review an unprotected program before I'll buy or review a protected one. But refusing to look at or use any copy-protected software in a world where novice users expect to get all the software they need for free is pushing logic too hard. (Experienced users, who have a better appreciation of what it takes to write a good program, appear to be much more likely to pay for their software.) I want to see an end to copy protection because it's a nuisance to honest users. But the only way to achieve progress on the copy protection front is to achieve progress on the copies-for-free front. It takes only a glimpse of holiday reality to realize that little progress is being made there. With no copy protection at all, Christmas could become nothing more than a time for families and friends to get together to cheat software developers out of honest livings. Many novice users actually seem to believe that if a program isn't copy protected it must be public domain. We have to get rid of those ideas. Until we do, honest users will simply have to learn how to live with the developers who insist on using copy protection. That's not, after all, impossible.

Ask (or tell) Uncle DOS

*Just when they finally had a good excuse to use an exclamation point, the gang at **Nibble** ran out of them! Look at the cover of their February issue! "Interview with Tom Weishaar, Publisher of **Open-Apple**" Geez. For more on this controversy, see "Miscellanea" in June 1985, page 1.45, and "**Nibble** bytes tongue," in August 1985, page 1.63. In the interview I'm quoted as saying once again that I'd like to see AppleWorks in ROM on new Apple IIs. I'm catching a lot of grief about that one; everyone says it would be too hard to patch that way and new versions couldn't be used. Sure would be a big sale for Claris, though.*

I thought we'd carried this correction before, but I can't find it anywhere. On April 1987's page 3.22, in the program listing in the middle column, in the line that begins 00/030A, the number after the colon should be A2, not AD.

***IIgs system disk 3.1 follow up.** Thanks to several postal and GEnie correspondents, I've learned a few additional things about the new IIgs system disk the last couple of weeks. Please thank them when you next see them. Meanwhile, pencil in a note on page 3.90 of last month's issue that says to look here for additional information when you buy that new IIgs.*

*First, let's set the record straight about the disk-drive light-blinking many of you have been worrying about while running the **Finder.** This is normal. The only way the **Finder** can tell if you've taken a disk out of a 3.5 drive is to repeatedly ask it. But each time a UniDisk is asked for its current status its in-use light blinks. Don't worry about it.*

*Next let's talk a bit about icons. Icons are those pictures that show up on your screen when you run the **Finder.** The idea is that each file type should have its own icon. Thus, you can tell a binary file from a text file from an Applesoft file by looking at its picture. In addition, icons can be defined for specific files. For example, a special icon could be created for an Applesoft program called HEARTBEAT.MATH. Just that one program would be linked to the special icon; other Applesoft programs would continue to use the generic Applesoft icon.*

*All icons that the **Finder** uses come from special files that are themselves file type $CA and that have a file name ending in ".ICONS" (no, I don't know what the icon for an icon file is). In addition, all icon files must appear inside a folder (previously known as a subdirectory) called ICONS.*

*The IIgs system disk has an ICONS folder in its root directory. In January, I insisted rather strongly that this folder contained the same kind of stuff as a bunch of other folders on the system disk and that those other folders were inside the SYSTEM folder. Thus, using blind logic, I insisted ICONS should be inside the SYSTEM folder, too. At the time I assumed that all icons the **Finder** would ever use would be in the System Disk's ICON folder.*

*However, subscriber David Lyons points out that every disk you own can have its own ICONS folder. The **Finder** will look for an ICONS folder on all the disks it has access to. Since most disks don't have a SYSTEM folder to put the ICONS folder into, it makes more sense for ICONS to be in the root directory.*

*The file DIALOG.ICONS, which I gave a "what is this and why is it here?" notation in the directory listing on page 3.90, may or may not deserve the snotty treatment I gave it. It contains duplicates of some dialogue icons that are embedded inside the **Finder** program. The **Finder** doesn't use DIALOG.ICONS, but I supposed it's possible that other ProDOS 16 programs could. But even so, why is this file cluttering up the root directory instead of being in the System Disk ICONS folder?*

*The file called BASIC.LAUNCHER in the root directory of the IIgs system disk, on the other hand, has value. Don't drag it to the trash can as I suggested on page 3.92. Don't believe my notation in the aforementioned directory listing that "it takes longer to load than to run." BASIC.LAUNCHER is needed to startup Applesoft programs from within the **Finder.***

*If you double click on an Applesoft program icon, the **Finder** is supposed to run that program. To get to that program, however, the **Finder** has to have a way to leave ProDOS 16, startup ProDOS 8, startup Basic.system, and, finally, startup the Applesoft program you have selected. This requires a long reach. The **Finder** obtains this reach by telling BASIC.LAUNCHER what Applesoft program you want to run, and telling PQUIT to start BASIC.LAUNCHER. This is how we get out of ProDOS 16 and into ProDOS 8. BASIC.LAUNCHER, in turn, loads Basic.system, embeds the name of the Applesoft program you want to run in the STARTUP position, and runs Basic.system. As you can see, without BASIC.LAUNCHER, you can't start Applesoft programs from the **Finder.** You can startup Basic.system from the **Finder** by clicking on the Basic.system icon, but you can't tell Basic.system to run anything other than the default program (STARTUP) without BASIC.LAUNCHER.*

*Interestingly, the **Finder** knows to use BASIC .LAUNCHER with Applesoft programs because "BASIC.LAUNCHER" is written in tiny print inside the icon file. The icon for binary files, on the other hand, doesn't include this information. That's why you can't start up a binary file from inside the **Finder.** However, using a buggy Apple program called **Icon Editor,** you can add the information about BASIC.LAUNCHER to the binary file icon. Neat, eh? Then novice users could crash their systems whenever they wanted by clicking on binary files that weren't actually programs. The same trick would allow you to EXEC text* *files from the **Finder**. Apple handed out a beta copy of its **Icon Editor** to Apple II developers at AppleFest, but, unfortunately, this tool hasn't surfaced at the Apple Programmers and Developers Association yet.*

*(Actually, icons are even neater than this. You can make an icon specifically for all text files that have a filename ending in ".EX," for example. If all such files are execable and you tell the icon to use BASIC .LAUNCHER, then you **can** exec these files from the Finder without endangering novice users.)*

*While we're talking about the **Finder**, I'll pass on a tip from GEnie user Bruce Gordon. Program icons can be dragged onto the desktop and left there. Run one of the programs (with the "write Finder data" option on), and when you quit back to the **Finder**, the icons will reappear on your desktop (if the programs can be found in your drives). You can double-click on any program's desktop icon to run it without opening any windows or folders. This trick should be most useful for hard disk users. Don't bother teaching it to your Macintosh friends; however, because their **Finder** can't do that unless all the programs come from the same window to begin with.*

*A **Finder** limitation several people have reported running into is that it won't copy the contents of a 3.5 disk onto /RAM5. As GEnie's LINEFEED (Lou Flemal) explains it, "the **Finder** uses the free RAM in the GS Ram card to copy faster. But you are trying to use that same RAM to copy TO. When **Finder** starts out, it sees xxxK free, so it knows it can fit the disk you want copied into /RAM5. But then **Finder** loads all it can into the free RAM (same stuff) so it can copy faster. When it goes to write the stuff to /RAM5, there isn't enough room, because it has used the space."*

*I found two notes about the system disk in Apple's AppleLink technical library. One says that anyone who upgrades to the 1.1 IIgs ROM should then only use the system software on system disks 2.0 or 3.1. The other says that the **Finder** "has a practical limit in the number of files per folder it can handle. Though having 200 or more files in a folder will not cause a system crash, the **Finder** slows to where even the simplest of operations becomes unacceptably time consuming." Dennis has demonstrated this has nothing to do with the **Finder** itself, however, but happens to all programs using ProDOS. For example, try this, and watch how slow things get as we approach the 200th file:*

```
10 PRINT CHR$(4);"CREATE BIG.DIR"
20 PRINT CHR$(4);"PREFIX BIG.DIR"
30 FOR I=1 TO 200
40 : PRINT "Now on file ";I;"."
50 : PRINT CHR$(4);"SAVE FILE.";I
60 : NEXT
```

The slow speed results from ProDOS thrashing around loading the same old directory blocks in and out of buffers. The moral: use more folders (subdirectories) so that you don't have more than about 50 files in any single folder.

*Subscriber Steve Busey points out that you can set up a two-volume hard drive so that the first volume (/HARD1, for example) boots into ProDOS 8, while the second volume (/HARD2, for example) contains the ProDOS 16 system files. To startup ProDOS 16, you simply set the prefix to /HARD2 and execute PRODOS. You can easily do this from a program selector such as **ProSEL**, if you want.*

*Subscriber Wilber Mardis points out that once you've created a system disk, you can copy many ProDOS 16 application programs onto 5.25 floppies. Once you get all the tools, fonts, and print drivers onto the system disk, older floppies often have enough room for the actual programs. "I have, for instance, copied **MultiScribe GS** to one side of a 5.25 disk and **Top-Draw** to the other."*

Getting down to business

I've read comments in your publication about using Apple IIs in "commercial productivity" applications. Let me tell you about how we do it.

We utilize eleven enhanced Apple IIc machines–all using AppleWorks–to operate an electronic instrumentation business. We acquired our first machines in late 1984 and have taken advantage of improvements in AppleWorks itself (version 2.0), and in various hardware (Z-RAM Ultra III, UniDisk 3.5) and software add-ons (*PinPoint*, *Point-to-Point*, and *TimeOut*, for example). We're experimenting with some relatively new AppleWorks file-compatible accounting software, too. If a IIc fails, we have a couple of extras, so we just whup it out an' drag the broken one down to the local Apple boys for service.

Every single business day, we use word processing, spreadsheets, and data base operations, and accessory programs for communications and spell-checking, exchanging disks and files, filling in standard order and invoice forms, making mailing labels, using information from our IIcs. We have a portable IIc (*a la* Roger Coats, in San Diego) utilizing a Prairie Power battery pack, an LCD display, a Diconix battery-powered printer and a ProCom modem, which we regularly carry to our Aberdeen, Scotland office to communicate with the home office network. We plan to put a laser printer on line this month. Several employees have personal IIcs at home, which they readily use to do the "home work" regularly done by successful businessmen today (although several of our guys admit that they have to share the home machine with their families). My daughter at college regularly exchanges files with the business network and with our home machine using her enhanced IIc.

We've solved lots of minor software and hardware problems to make our system work and grow, but Apple support people–local and regional–were, and are, *no help at all*. It's a real pity that other businesses who would like to do what we do (and save the money that we've saved) are shunted off to Macintosh-land by the locals, who steadfastly deny that a IIc can be productive. And, worse than that, many of these fine potential II users become convinced by others that only MS-DOS'll do it, and we don't hear much from them anymore (yes, we do have one MS-DOS unit in the corner, and, yes, it'll soon be connected to the network as well). We know the limitations of the IIc CPU and graphics, but let's face it–those limitations mean very little in actual, real daily business transactions.

Now, don't get me wrong. The local Apple folks didn't actually *fight* with us as we bought, or inquired about, the various components we needed. They just said "We don't know about that," nearly every time we asked a question. It was too much to expect the Apple locals to know about enhancing AppleWorks, or enhancing IIcs, or doing a bit of networking, I guess.

After a bit, we, like other slightly knowledgeable Apple business users, bought our goodies by mail order (we succumbed to the "Why pay more from someone who won't help you?" syndrome), causing the locals to decide that there wasn't enough Apple II business around to warrant their attention, as they leaned further toward MS-DOS contraptions. It's a doggone shame that we businessmen must also be computer experts, and work *against* the local Apple sales effort to get the job done, while the MS-DOS boys down the street will gladly sell you the whole lot.

Our business network operates just fine. We don't brag about it–we just *use* it. It's nothing fancy, but 11 machines with compatible, integrated software that is easy for new employees to learn and a varied collection of accessible printers–Apple and otherwise–is nothing to sneeze at, my friend, Macintosh, MS-DOS, or otherwise.

To top things off, we developed and built a cheap standby rechargeable power pack that operates each IIc for 2 hours during power glitches and outages. It lets us really utilize IIc RAMcards and RAMdisks. It costs 30 bucks to make, and any Apple IIc user can get all the parts at Radio Shack. Some folks sell them for as much as $169. This little gadget, if sold or marketed by Apple, would allow more folks like us to use IIcs as solid productivity computers. If anyone would like a Radio Shack parts list and schematic for our IIc standby power unit, just drop a SASE to us, and we'll fix 'em up.

How many more businesses operate Apple II office systems like we do? Why has Apple deliberately chosen to ignore us, and our needs? Why can't we get info on how to link IIcs together, or to laser printers, from the local Apple boys? We've been tempted to find, or start, an Apple II "business only" users group. Maybe there is such an organization, and we're further behind than I thought. Oh, well, back to exchanging files and graphics with Scotland, as we prepare our 400K one year operating plan spreadsheet, and print it out on the laser printer as the Customer Service guys run a database failure sort for a customer who has asked Marketing for a big re-quote, while Engineering fiddles with...

George Smith, President
Electro-Flow Controls, Inc.
P.O. Box 870
Missouri City, TX 77459

Useful computing, ordinary people

I have often pondered why the Apple II has such appeal to me after using a II-Plus then a IIe. It has been very difficult to actually pin down the reasons for my loyalty. Certainly much has to do with accumulated knowledge but there is much more to it than that. Probably the major factor, and one which goes against conventional wisdom in the computer world, it that "simpler can be better." This leaves me feeling in control of my computers (one at home and one at work) and not feeling overwhelmed and useless when something unpredictable happens.

I regard myself as a serious user. I work in the area of youth policy and often write reasonably lengthy documents, a regular newsletter, and a welter of short documents as well as maintaining a number of smallish databases. I use my computer every day at work and it really is a personal productivity tool rather than an embellishment on my desk.

It strikes me that many people writing about computers are people who write about nothing but computers. As such they are all too infatuated with "new" and "powerful" and "fast" and "complex" and so on. This is wonderful if you have nothing to do all day but play with the latest 386 monster and explore the intricacies of OS/2 but what about us folks who use a computer as a means to an end rather than as the end? Similarly, I find many people using MS-DOS computers equally hung up on these marketing concepts and not actually doing much at all with that power and complex software. The price they pay is a most unfriendly operating system, a plethora of standards (which DOS versions?....which graphics standard? ....what format word processor files?), and noisy fans.

If there is one thing than an Apple (and AppleWorks) can do well, it is make useful (rather than esoteric) computing available to ordinary people.

I'd also like to endorse the comments of every second writer to ***Open-Apple*** and lament the lack of knowledge of dealers. They are no better here than anywhere else. I recently had to tell one about the IIe to IIgs motherboard upgrade option among a number of other things.

A possible solution to the chronically inadequate support provided by many dealers would be for them to enlist local Apple II aficionados as "experienced customer advisors" to speak to potential or actual purchasers about the possibilities of their computers. Something as simple as giving a potential customer the name of an experienced local user as a support person would be of enormous value to the dealer; the local support person could receive special discounts as payment. Every happy user is worth a great deal to Apple. Conversely, every unhappy user who puts their system away in frustration is a mini-disaster. I, for one, would be quite happy to be involved in such a scheme.

Ian Wright
Ballarat, Australia

Stone Solid Apple

I've been reading your comments and letters about the future of the Apple II with interest. Where will new Apple II productivity software come from? From fools like me who still believe in the Apple II!

Fools who believe that if you can write a decent database manager for a 48K computer, you should be able to write a *great* database manager when you have 128K to work with.

Who believe that people are interested in fast, powerful programs that don't use graphics and sound, even if Apple Computer generally isn't.

Who love the Macintosh *and* the Apple II, but suspect that trying to turn an Apple II into a color Macintosh simply costs too much in *both* speed and memory to make sense for a lot of applications.

Who are willing to buck all of the trends and go it alone, without big-bucks financing (the way we did it in the "old days", when personal computer software was a vibrant, innovative cottage industry) producing high-powered applications software for a computer that "conventional wisdom" says won't pay off.

Who understands that first time computer users soon become experienced computer users, with needs that often can't be met by software designed for first time users. (But they're still computer *users*, not computer *programmers*. And a good program designer should be able to give those people the power they need without forcing them to *become* programmers.)

Fools who are willing not only to produce software for the Apple II, but to make a *commitment* to it.

Stone Edge Technologies is committed to making *DB Master Version Five* and *Version Five Professional* the high-power data base managers for the Apple II. With multi-file relational capabilities, multi-user versions, and one of the most powerful report generators available for any personal computer, we're giving the Apple II market a chance to prove that the pundits are wrong–that there *is* a market for high-powered software for the Apple II.

Barney Stone
Stone Edge Technologies, Inc.
P. O. Box 200
Maple Glen, PA 19002

Blessed are the disk rescuers

You mentioned Pete Johnson as a source of disk repair in an earlier issue (November 1986, page 2.76). I have had, unfortunately, several occasions to need his services. He has always done well for me. The last time, I needed help FAST. I express-mailed the disk to him; I had it back the third day. I heartily recommend him.

Brian Eastman
Cincinnati, Ohio

Johnson's address is P.O. Box 5, New London, MN 56273.

Solving for double-blanks

A simple way to get around the "calculated double-blank @NA" problem mentioned in "*Multiplan* update" in December 1987, page 3.85, is to create a duplicate of what you want to print on the right side or the bottom of your work area. To create this duplicate, where, for example, cell A101 corresponds to the original A1, use the formula @IF(A1=0,@NA,A1). Using the copy/relative command it takes just a few keystrokes to create this duplicate spreadsheet, provided you have enough space.

I found in practice that I want the cosmetics of blanks substituting for zeros only in that part of the spreadsheet that I print into a report. And that is usually only the summary. The bigger the part with the cosmetics, the slower will be the speed of calculation. That's why I try to keep the cosmetic part as small as possible.

Another possibility, when there are just a few columns or rows that contain calculated double blanks, is to use the @IF function to determine where the blanks are. For example to add A10 and A11 when either could contain a double blank, use @IF(A10=0,A11,@IF(A11=0,A10,A10+A11).

Werner Kubelka
Niteroi, RJ, Brazil

Use a "flag cell" to blank zero entries in worksheets (December 1987, page 3.85; June 1987, page 3.39). Set the flag to 99 to see the spreadsheet as it would normally appear and to calculate totals. Set the flag to zero and recalculate to blank out the zero cells for printing.

In the data cells, use @IF with a formula such as @IF(@AND(flag=0,result=0),@NA,result), where "flag" points to the cell you will use as a flag and "result" is the formula that gives the data you want displayed in this cell.

In cells that make calculations using these data cells, use a formula such as @IF(flag=0,self,calculation), where "self" causes the cell to return its own value and "calculation" returns the data you want displayed in this cell. Force calculation of the worksheet with the flag set to 99 to obtain the correct results, then set the flag to 0 and recalculate to replace the zero cells with blanks.

For installing custom printers, a word processor file of printer code macros would be an efficient distribution medium. The macro could start at the point where you give the printer a name.

Has anyone noticed that Randy Brandt is licensing a runtime version of *UltraMacros* to support task files?

You published a letter of mine in September ("Thinking about databases," page 3.63) in which I said the AppleWorks word processor Find and Replace commands take about 10 per cent longer with version 2.0 than with version 1.3. My AppleWorks 2.0 word processor was slower because it had escaped my permutation heap unlabeled with *AutoWorks* on it. In reality, the 2.0 word processor is slightly faster.

I second the motion for a discussion of what a 16-bit (or any) AppleWorks could and should do. For example, version 2.0 still cannot handle a DIF file with optional headers, a bug you exposed in detail in August 1986 ("AppleWorks DIFficulties," page 2.56).

Robert Ericson
North Providence, RI

Along with his letter, Ericson sent us a copy of a new, second edition of his book ***AppleWorks Tips and Techniques*** *(Sybex). I liked the first edition of this book so much (January 1986, page 1.97) that part of what I said ended up on the back cover of the second edition. The new edition was just generally spiffed up with new tricks, expanded to include all the new stuff in AppleWorks 2.0, and has a whole new chapter on macro programs. Here's a typical Ericson insight from the macro chapter, "The Beagle Bros manual draws no line between macros and control programs. This makes macros appear to be more complicated than they really are."*

If you buy just one book about AppleWorks, this is the one to get.

AppleWorks, 8 & 16

Your ongoing discussion of what the next version of AppleWorks should do is intriguing. However, the focus of the discussion has been on the IIgs and what a 16-bit AppleWorks should be like. Why not bring any enhanced version of AppleWorks to the 8-bit IIe and IIc too?

Both 8-bit and 16-bit versions of AppleWorks should be included in the same box. The 16-bit version should use the 16-bit code for improved performance, not new features, as you have suggested. To the user, and to AppleWorks add-ons like *Timeout*, both versions should seem identical. To allow add-ons to be compatible with both versions, the developer hooks should be written in 8-bit code.

Now, how should the program itself be improved? For one thing, the three AppleWorks applications should be enhanced to bring them up to stand-alone quality. In the word processor, for example, how about right justification, a way to print either formatted or unformatted text files without using a custom printer, and wildcard and carriage return options for Find and Replace?

A way to embed special printer codes in word processor documents is needed. Why not allow ten custom character strings per printer? In the word processor, select U(ser)0 through U9 to select these strings, which would appear as carets in the text. These commands would be sent to the printer (but not counted in the line length) during the printing process.

The data base needs more flexible reporting capabilities and the spreadsheet needs more functions and string-handling ability. As Lowney suggested, a pathname selection menu should be included and should be invokable any time a disk is accessed. AppleWorks should display subdirectories when loading a file and allow the user to open them and look inside. Text and DIF files should be loaded from a menu. Mouse support, in the AppleWorks spirit, that does not replace the standard AppleWorks interface, would make AppleWorks simpler for new users.

AppleWorks should recognize and use any combination of aux-slot, standard slot, and IIgs memory, but should be smart enough to leave RAMdisks alone. We need more word processor lines, data base records, spreadsheet cells, and a larger clipboard. The desktop should be able to hold a larger number of files.

Finally, AppleWorks needs something similar to Beagle Bros' *Timeout* manager built in, to support applications that run within AppleWorks. Add-on programs should be able to access every necessary AppleWorks I/O, disk, and memory feature. New applications should be able to appear on the Add Files to Desktop Menu and should be activated when you load a particular file type. And, of course, AppleWorks developer's information should be made available in the form of a technical reference manual.

Jerry E. Kindall
Grove City, Ohio

75-file bug

I have all of my AppleWorks files carefully arranged in subdirectories on a Sider 10 meg drive. Some of them are getting rather full. One of them, /AW/DATA/LETTERS/COMPUTER, currently has around 75 files. When I attempt to load any of them into AppleWorks 2.0 on my IIgs, everything proceeds fine until the last file specified is loaded. AppleWorks then tells me it is getting errors trying to read "" (empty double quotes). At the bottom of the screen is the "Press spacebar to proceed" message.

Here's where it gets weird. If I press the space bar, the program crashes hard. So hard that a *Timeout* control-reset will not recover the program. But,...if I press Escape instead of the space bar, I go back to the main menu and *all* of the files I have specified have been properly loaded. This happens even if I specify only one file.

I have done some tests to try to figure out the source of the trouble. I've tried several copies of AppleWorks 2.0 under several versions of ProDOS. Always the same. However, under AppleWorks 1.3 I never have a problem.

Rex Creekmur
Grandville, Mich.

We can't explain it either. We've had reports that problems like this happen with as few as 64 files in a subdirectory (July 1987, page 3.45) and reports that AppleWorks 2.0 is designed to handle a maximum of 85 files in a subdirectory (August 1987, page 3.56). We once a carried a patch to expand this (also page 3.56), but it doesn't work, as I ruefully admitted on page 3.69. I suspect the only short-term answer is to put fewer files in each subdirectory. Since Claris has big plans for customer support, perhaps we can look for a fix to this problem from them.

Database fixer

I've also experienced the AppleWorks 2.0 database file bug reported by Eugene Whitehouse and Sidney Powers (December, page 3.87; January, page 3.96). I don't know what's causing it, but I've gotten quite good at recovering the files. If *Copy II Plus* is used to copy a damaged file to an otherwise empty disk, the database file header will be found at track 0, sector 1. Byte 38 ($26) in this sector gives the number of report formats in the file. I've had success at both trial and error changes to the value and at studying the first 50 lines or so of a word processor "text" file load of the database file to determine just how many report formats are left.

Robert L. Hildebrandt
Pleasant Ridge, Mich.

*Those who would rather use **ProSEL's Block.Warden** to make the change need to tell it to F(ollow) the correct file, then change byte $26 in the file's first block.*

Another copy machine

Another way to make more than nine copies of a word processor document without having to patch anything ("AppleWorks as copy machine," October 1987, page 3.71) is to create a database file with just one category. Put a single word from your document in this category, then copy the record until you have as many records in the database as you want copies of your word processing document. Use mail merge to insert that word back into the document from the database. AppleWorks 2.0 will print the exact number of copies that you want.

Larry Jones
El Paso, Texas

ADB.READER bug found

There's a bug in your ADB.READER program in the March and June 1987 issues. AppleWorks database records begin with a two-byte length. When a record starts in the very last byte of the buffer, however, ADB.READER gets only the first byte of the two-byte length. The second byte is whatever garbage happens to be one byte past the end of the buffer. To fix this, the end of buffer pointer (BEN) should be aimed at the last byte of the buffer rather than one byte beyond the end of the buffer. Change line 1052 on pages 3.11 and 3.35 to:

```
1052 BEN = BBG + 16384 - 1 : REM BEN=end of buffer.
```

In March's line 1052 you divided the 16384 by 2; in June you said this wasn't necessary, so I didn't do it here.

Jim Luther
Kansas City, Mo.

A sneaky right-justify

Here's how to get page numbers (or any other text) right justified inside an AppleWorks header. Use open-apple-O(ptions) HE to start a page header, and JU to force full justification. Enter the text to be right justified, a blank space, and then about 10 sticky-spaces (open-apple-space bar). If you want the rest of your text to be unjustified, return to open-apple-(O)ptions and enter UJ. Now, put the insert cursor at the beginning of your header text and press the space bar until the sticky spaces drop to the second line of the header. Your AppleWorks screen should now look something like this:

```
--------Page Header
--------Justified
                                        page ^
^^^^^^^^^^
--------Unjustified
```

The caret after the word "page" is the page number caret, the other carets are sticky-spaces. AppleWorks allows only one line of text in a header. The idea here is to push the sticky spaces into the second line, where they won't be printed. But they will drag the page number to the right margin because the header is being printed with full justification. This right justifies the page number even if it grows to a two- or three-digit number.

If you are using one of the proportional typestyles, you may have to use more spaces in front of the text (until the text is about halfway across the second line) and more sticky spaces to get the trick to work. This is because the space is a very narrow character with proportional typestyles.

You can also use this trick outside of headers–for example, in a table of contents–but in this case the sticky spaces will actually be printed as a blank line. So you have to plan your printout so that it appears the output is (at the least) double spaced, because every line following a right-justified line must be blank.

Hans-Juergen Kuehne
Garbsen, West Germany

AppleWorks passwords

I was wondering about the possibility of patching AppleWorks 2.0 to add a password routine to prevent casual accessing of sensitive files. This valuable feature is available on *Symphony* and it would be a good addition to AppleWorks.

John M. Klein
Spring, Texas

*The best solution we can think of is Beagle Bros **Timeout DeskTools**, which contains a routine called **File Encrypter** that can be used to encrypt files from within AppleWorks. You load any type of AppleWorks file you want to keep secret onto your desktop, encrypt it using a password of your choice from one to ten characters long, and then save it. Encrypted files look like gibberish. To decrypt, load the file onto the desktop, reenter your password, and the file will be restored. If the file is still gibberish, you entered the wrong password. At this point you can try to guess the right password, but it may take some time to enter all 60,000,000,000,000,000,000 possible combinations.*

***Desktools** also includes a calendar, calculator, notepad, dialer, envelope addresser, clock, clipboard converter (for moving data directly between database format and spreadsheet format, or vice versa, without using DIF files), upper/lower case converter, word processor page preview, word counter, and, of course, a puzzle. **Timeout DeskTools** is $49.95 from Beagle Bros, 6215 Ferris Square, San Diego, CA 92121 619-452-5500.*

aedi revelc A

I was recently approached by an associate who operates a small business. He has a number of data disks that are vulnerable to unauthorized copying or use. He wanted me to assist him in the selection of an encryption system.

After considering his request, I decided to simply reverse the direction of the drive motor in his second drive, which is where he always puts his data disks. Once the direction change was made, we used *Copy II-Plus* to copy his disks from the normal drive to the reversed drive. The system and software don't know that a change was made. But the disks made on the reversed drive can't be read by a normal drive. The major benefit of this encryption system is that the user has no added steps, pre-boots, or operator intervention, so there's no clue that something is different. I put a switch inside the drive, operable through a hole in the bottom, so that normal operation of the drive can be achieved without disassembly.

To make this modification, take a drive apart and locate the drive motor. There are normally four wires connected to the motor, two for DC power and two for speed control. Disconnect one of the pairs and see if the drive motor will come on. If it does, you've disconnected the speed control wires and you should reconnect them.

Once you've figured out which pair of wires provide the DC power, look at the back of a DPDT switch and you'll see six places where wires can be connected, three on the right side and three on the left side. Solder the power lines from the circuit board to the two top connectors, one to the right and one to the left. Also attach a wire from the top right connector to the bottom left connector and from the top left connector to the bottom right connector. Finally, solder new wires going to the drive motor to the two middle connectors, one on the right side and one on the left side. With the switch in one position, the drive runs normally, in the second position it is reversed. Move the switch only when the drive motor is off.

Len Powell
Finksburg, Md

ProDOS time

I want to access the Apple IIgs clock from Applesoft under ProDOS. I tried PEEKing at locations 49042 ($BF92, minutes) and 49043 ($BF93, hours), but these locations are only updated *during disk access*. Do I need a machine language routine to call the IIgs toolbox, such as the one you published back in April (page 3.21), or is there an easier way?

Glenn Calderone
Fountain Valley, Calif.

The easy way is to use the ProDOS Machine Language Interface GET_TIME call. This call updates the locations you mention with the current time. Using this call, rather than a IIgs toolbox call, makes your program compatible with any Apple II.

Here's a short program that Dennis wrote to show how to do this:

```
10 POKE 768,32  : REM 0300:20 00 BF JSR $BF00
20 POKE 769,0   : REM 0303:82        .DA #$82
30 POKE 770,191 : REM 0304:00 00     .DA $0000
40 POKE 771,130 : REM 0305:18        CLC
```

```
 50 POKE 772,0   : REM 0306:60       RTS
 60 POKE 773,0   : REM
 70 POKE 774,24  : REM $82 is GET_TIME call #
 80 POKE 775,96  : REM   no parameter list
 90 CALL 768
100 PRINT "Time is ";PEEK(49043);":";PEEK(49042)
```

Printing pastels

I found this hint quite by accident, but now I use it any time I want to print pastels, rather than vibrant colors, with my ImageWriter II. Simply set the printer's paper thickness control for one additional sheet of paper. This makes the printer print lighter than it normally does, something you might want to use with a baby announcement or shower invitation. The one problem is that black won't be as black as it should be.

Does everyone know you can return to the *Finder* from the Applesoft prompt on the IIgs by just typing "BYE"?

Lowell Lutz
Phoenix, Ariz.

Hello Good Bye

I have just received a copy of ProDOS 8, version 1.4 on a commercial program disk. I would like to know of a standard way to install Alan Bird's *Bird's Better Bye* quit routine into the various versions of ProDOS 8.

H. W. Madison
Bellingham, Wash.

***Bird's Better Bye** will work with any version of ProDOS 8. You need to lift a copy of it out of your current version, lay it into the new version, then BSAVE the new version with Bird's program embedded in it.*

Here are the details you need to know to do this. The program has a length of $300. In versions 1.2 through 1.4 of ProDOS, it lives at $5900 if you bload PRODOS at $2000. In versions previous to 1.2 it lives at $5700. So, for example, to move the program from a version 1.1.1 PRODOS file to a version 1.4 PRODOS file, do this:

```
insert disk with PRODOS 1.1.1
BLOAD PRODOS,A$2000,TSYS        (load ProDOS 1.1.1)

insert a backup copy of a disk holding PRODOS 1.4
BSAVE BIRDS.BYE,A$5700,L$300  (save quit code)
BLOAD PRODOS,A$2000,TSYS        (load ProDOS 1.4)
BLOAD BIRDS.BYE,A$5900          (overlay Bye)
BSAVE PRODOS,A$2000,TSYS        (re-SAVE ProDOS 1.4)
```

Bird's program isn't sold separately. It's part of the ProDOS image on most Software Touch and Beagle Bros products.

Duplicate pathname problem

Copy II Plus users, beware! The program will allow you to rename a file to that of one which already exists. If you then try to manipulate the renamed file and it happens to be farther down in the directory than its namesake, you'll be manipulating the wrong file.

EXEC files make a good alternative to the Basic Copy program in your July 1985 issue. In my application, the last thing that my STARTUP program needs to do is to transfer some files to /RAM. I put all the LOAD/SAVE's, BLOAD/BSAVE's, and the RUN MAIN.PROGRAM into a text file (with my word processor), save this as a text file, and then have my startup program EXEC the filename and END.

To overcome the annoyance of an "]" being printed for every command executed, POKE 768,216 and 769,96 then do a PRINT D$;"PR#A768". What you get is a short routine (CLD, RTS) that will consume all characters sent through the output hook. (The CLD is required by ProDOS.) My main program resets output with a PRINT D$;"PR#3".

Paul Lucas
Levittown, N.Y.

ZBasic amplifications

You got a couple of points wrong about *ZBasic* in your reply to "Basic choices" in December, page 3.88.

You said our DOS 3.3 version runs on any 64K Apple. While compiled DOS 3.3 *ZBasic* programs will run on any 64K Apple, our DOS 3.3 version requires a 128K IIe, IIc, or IIgs for program development.

Our ProDOS version comes in two flavors, 64K and 128K. The 64K version will run on any Apple that has at least 64K (this is required for ProDOS), as will the programs compiled with it. The 128K version requires at least 128K and a 65C02 or newer processor. In addition, any program compiled using the 128K version also requires 128K and a 65C02.

When a compiled program is distributed to others, the *ZBasic* runtime support files may accompany the stand-alone file without royalty payments or credit to Zedcor. However, the Editor, Compiler, and associated files cannot be legally distributed as your article implied.

Greg Branche
Zedcor
Tucson, AZ

On rereading my response I find I did imply that the whole package could be redistributed, but I didn't mean to. My goal was to compare the cost of actually distributing programs written with each of the various Basics.

G(uess) S(yntax) BASIC?

In the November **Open-Apple** you mentioned Apple's *IIgs BASIC* (page 3.80).

I have had it for a month and have two problems with it. First, there seems to be no way to warm start it. Control-reset brings up a bouncing Apple screen.

Second, LISTing a program to an ImageWriter II seems impossible. I can LIST to the screen, or print the output of a program or a CATALOG, but I can't OPEN .PRINTER as a file in immediate mode. In deferred mode, the error "RESERVED WORD" comes up when LIST is used in a program.

R. C. Allison
Punta Gorda, Fla.

*Dennis responds: The beta version **IIgs BASIC** manual is woefully deficient; we've had a copy for a while, too, but have decided not to review it until a passable manual surfaces.*

An example of the manual's problems is that file and device I/O isn't covered except in the reference section descriptions. This means you have to discover things in bits and pieces. I found one clue for listing a program under the syntax for the "OUTPUT#" command. Then I found out that the syntax given for the OPEN command is incorrect when opening a device. The example given is:

```
OPEN .CONSOLE, FOR OUTPUT AS #1
```

The correct syntax includes quotes around the "filename":

```
OPEN ".CONSOLE", FOR OUTPUT AS #1
```

That solved, I scanned the GSBASIC.SYS16 program file with Block.Warden to find out what the other (unspecified) device names were; I found CONSOLE, PRINTER, MODEM, NETPTR1, NULL, and MEMBUFR. NULL is probably a dummy device for sending output off into the ether, NETPTR1 is probably for AppleTalk devices, and MEMBUFR is described in the GS-BASIC manual as a "256 byte pseudo-character device" (Tom says they must mean a "256-byte pseudo character-device;" he says a pseudo-character would have to be a character with the high bit set to two).

Anyway, you can get an immediate mode printer listing like this:

```
OPEN "PRINTER", FOR OUTPUT AS#1
OUTPUT#1 : LIST : OUTPUT#0
CLOSE#1
```

This is pretty klunky; what's more, I haven't figured out how to get rid of a superfluous linefeed as each line "wraps" at the printer. You may want to enter the commands into an EXEC file; I find typing "EXEC LIST" is a lot easier than entering the above every time I want a hardcopy.

*I also haven't figured out how to use LIST from within a program. This feature may seem useless, but, for example, several of the Applesoft programs on the **DOStalk Scrapbook Program Disk** LIST themselves so that the user can see them and learn how they work.*

The answer to the control-reset situation is trivial. Just never include any bugs in your program that would make the program hang. If you do this, you'll never need control-reset.

Open-Apple

is written, edited, published, and

© Copyright 1988 by Tom Weishaar

with help from

Tom Vanderpool **Sally Dwyer**
Dennis Doms **Steve Kelly**

Most rights reserved. All *programs* published in **Open-Apple** are public domain and may be copied and distributed without charge. Apple user groups and significant others may obtain permission to reprint *articles* from time to time by specific written request.

Open-Apple has been published monthly since January 1985. World-wide 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; bound, indexed editions of our first three volumes are $14.95 each. Volumes end with the January issue; an index for the prior volume is included with the February issue.

Please send all correspondence to:

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

Open-Apple is available on disk for speech synthesizer users from Speech Enterprises, P.O. Box 7986, Houston, Texas (713-461-1666).

Open-Apple is sold in an unprotected format for your convenience. You are encouraged to make back-up archival copies or easy-to-read enlarged copies for your own use without charge. You may also copy **Open-Apple** for distribution to others. The distribution fee is 15 cents per pege per copy distributed.

WARRANTY AND LIMITATION OF LIABILITY. I warrant that most of the information in **Open-Apple** is useful and correct, although drivel and mistakes are included from time to time, usually unintentionally. Unsatisfied subscribers may cancel their subscription at any time and receive a full refund of their last subscription payment. The unfilled portion of any paid subscription will be refunded even to satisfied subscribers upon request. MY LIABILITY FOR ERRORS AND OMISSIONS IS LIMITED TO THIS PUBLICATION'S PURCHASE PRICE. In no casa shall I or my contributors be liabla for any incidental or consequential damages, nor for ANY damages in excess of the fees paid by a subscriber.

ISSN 0885-4017 **GEnie mail: OPEN-APPLE**
Printed in the U.S.A.