

A journal and exchange of Apple II discoveries

Graphically speaking

by Bo Monroe

Since this article is going to examine some of the new software that's making the Apple IIs more colorful, let's start the proceedings by looking at what a "color" is on the IIs. Everything you see on the screen is made of red and/or green and/or blue light. Any color you care to see (and some that maybe you don't) can be mixed from some combination of these three primaries. Black? None at all. White? Red, green and blue. Yellow? Red and green. And so on.

The IIs is at heart a digital beast, so it figures that it does out primary colors in numerical quantities. It can send a primary color signal to the monitor in one of sixteen intensities (0-15). Zero means that the color is "off". Fifteen means it's on full blast. There are three primary colors, remember, so it follows that the computer can display sixteen times sixteen times sixteen, or 4,096 different colors. And indeed that's precisely the size of the Apple IIs's "color universe".

Pixel Maps: how the colors get to the screen

As you probably know, colors don't just appear anywhere on the screen. Instead, the screen is divided into thousands of discreet little points of light called "pixels". Each pixel can be lit with a single RGB color. ("Pixel" is short for PIXure ELEMENT — remember, the average computer weenie spells about as well as the average Our Gang Comedy character.) The pixels are arranged in a utilitarian, albeit boring, rectangular grid. The number and size of the pixels determine how fine or coarse the display is. The fineness or coarseness is usually referred to as "resolution" and is usually expressed as the number of horizontal pixels times the number of vertical pixels, sometimes with the "pitch" (number of pixels per inch) thrown in there somewhere. When you go into "Fat-bits" mode in your fave paint program, the big blocks of color represent individual pixels.

Each pixel on the screen is wired more or less directly to a small parcel of memory inside the computer. All the parcels put together form what is known as a "pixel map". The CPU stores numbers in these parcels; the number is converted into a color signal and sent along to the monitor. This should explain the psychedelic effects you see when your TransWarp overheats.

By now, perhaps, the binary-minded among us have deduced that each of these parcels must be at least twelve bits big (or "deep" in contemporary jargon), for twelve bits would be just enough to represent "4095" in binary. Wrong. Too bad, too, because if the IIs had "12-bit" color graphics, it could display any one of the colors in its universe at any pixel at any time. This would be nice indeed, for more colors per pixel means more than just a bigger box of crayons. My Mac's monitor, for instance, has only slightly greater resolution than that of my Apple IIs. However, it is capable of displaying color images of photographic quality because each pixel can choose between literally millions of colors. Lots of colors can increase the apparent resolution largely because not all parts of every object

depicted on the screen will fall squarely within the bounds of a pixel. Instead, some objects will fall only partly on a pixel. If enough colors are available, the pixel can be set to an average of all the colors that fall there. You can see this effect on the Mac by loading a suitable image and fiddling with the "Monitors" control panel. With the monitor set to two colors (black and white), the image is very coarse and abstract — even on a 640 x 480 screen. As you increase the number of colors to 16, the image is still coarse but now it's in color; the look is like something from a Marvel comic book. When you turn it up to 256 colors the image starts to look pretty realistic. By the time you get to 16.7 million colors, the image looks smooth and downright photographic — on the same old 640 by 480 screen.

But back to the real world. The Apple guys didn't give the Apple IIs "12-bit" graphics. Further proof that the IIs was intentionally crippled? Probably not. Increasing the bit-depth of a graphics display has some undesirable side effects. For one thing, the amount of raw memory required just to display the image increases. Graphic-oriented disk files increase in size, too. Because there's more information on the screen to handle, the CPU has to do a lot more work — and therefore the whole system slows down. My Mac (it's a fast one, too) starts to bog down noticeably as I turn up the colors, even if I'm working with a strictly black and white application. The Apple IIs engineers had to take these factors into consideration when they nailed down the machine's graphics capabilities.

In the end the guys behind the IIs settled on not one, but two compromise schemes for displaying graphics. The schemes are



"OKAY, NOW ENTER
AskFinderWhereAreMySocks."

known collectively as Super Hi Res, and are differentiated by their horizontal resolution. In Super Hi Res 320 mode, there are 320 pixels across the screen, and 200 from top to bottom. The four bit "memory parcel" assigned to each pixel can accommodate values between zero and 15. I won't bore you with the math, but 320 mode requires 32k of RAM for its screen pixel map.

320 mode pixels are almost, but not quite, square. They are slightly taller than they are wide — the exact amount depends on how your monitor is adjusted. 320 mode is not well suited for displaying gobs of text. The problem is there are not enough pixels across the screen to display anything but the chunkiest text face. Hence the second SHR mode. This alternate mode still has 200 pixels from top to bottom (a limitation dictated largely by affordable monitor technology), but with 640 pixels across. Now, the screen doesn't get any wider when the GS switches to 640 mode, so something has to give. That something is the pixel shape. "640 mode" pixels are distinctly rectangular — tall and skinny.

For who knows what reasons, the engineers decided to restrict 640 mode to the same 32k-size pixel map they allotted 320 mode. Something had to give here, as well. In this case it's the number of bits assigned to each pixel. 640 mode has twice as many pixels as 320, so it uses half as many bits per pixel — two.

Palettes. So how does the Apple IIgs squeeze 12 bits of RGB information into 2 or 4 bits? In other words, how does the screen know which 4 or 16 of the 4096 possible colors it is supposed to use? The answer has to do with tables of RGB color info called "palettes". An Apple IIgs palette has 16 such color definitions. The number stored in each pixel's memory parcel corresponds to one of the entries in the palette. For example, let's imagine that the second entry in the palette contains 3840 — the IIgs's RGB code for bright red. For every memory parcel that has two stored in it, the corresponding pixel will burn bright red. Now, if we put 4080 (RGBese for yellow) into the second entry in the palette, all the bright red pixels will suddenly turn yellow.

640 mode palettes. As we've already noted, 320 mode pixels have four bits assigned to them, allowing each to hold one of 16 values and thus providing access to any of the 16 colors in the palette. Though 640 mode pixels have but 2 bits assigned to them (and therefore can only hold one of four values), 640 mode palettes have 16 entries, just like their 320 mode counterparts. (Gee, Bo, that is odd.) The reason for this is the amazing phenomenon known as dithering. You see <pun>, 640 mode pixels are so skinny that the ol' eyeball tends to blend adjacent pixels together and perceive the average color. To take advantage of this, the engineers arranged things so that a 640 mode pixel finds its four colors within the palette based on its horizontal position on the screen. Thus, pixels in the first column use the first 4 colors in the palette, those in the second use the next group of 4 colors, and so on. The pattern repeats every 4 pixels. The resulting groups of colors within the palette are known as "minipalettes". Now, by carefully and cleverly arranging the colors in the minipalettes, it's possible to create the illusion of more than four colors available. For instance, in the standard 640 (or "default") palette, each minipalettes contains black and white. This means that any pixel can be black or white, and insures crisp black and white text. The other two entries in the minipalettes are up for grabs. Since dithering doesn't really work with more than two adjacent pixels the odd numbered minipalettes are identical, as are the even. The even pair of minipalettes contain red and green, while the odd pair has blue and yellow to fill out their entries. This yields the following two-pixel dithers:

Odd	Even	looks like:
Black +	Black	Black
Black	Blue	Dark Blue
Black	Yellow	Olive Green
Black	White	Medium Gray
Red	Black	Maroon
Red	Blue	Violet
Red	Yellow	Orange

Red	White	Rose
Green	Black	British Racing Green
Green	Blue	British Navy Sea Green
Green	Yellow	British Navy Pee Green
Green	White	British Navy Pea Green
White	Black	Medium Gray
White	Blue	Light Blue
White	Yellow	Pale Yellow
White	White	White

Provided you are depicting a beanery or some other verdant scene, the default 640 mode palette is pretty good. Otherwise, it pretty much stinks. And because colors are mixed indirectly with dithering it's usually impossible to come up a palette that is perfectly suited to the task at hand.

So in summary, 320 mode allows greater freedom of color, at the cost of resolution. 640 mode trades color flexibility for higher horizontal resolution. In practice, this has meant that graphic-intensive programs (like paint progs) usually stick to 320 mode, while text intensive applications use 640 (Appleworks GS). Programs that are neither fish nor fowl (HyperStudio) may opt to let the user switch between the two modes.

It's important to realize that 320 and 640 mode are essentially interchangeable. Both modes use the same basic data structure, but interpret it slightly differently. Pictures created in 640 mode can be displayed in 320 mode, and if the palette is set to mimic the 640 palette, the only thing you'll probably notice is text looks a little fuzzy. 320 mode pictures can likewise be viewed in 640; aside from the colors being totally screwy, the image will be perfectly legible.

But wait! There's more...

The Apple IIgs design team must have sensed we wouldn't be content with the limited color ranges of the SHR modes as described so far, so they threw in a bonus. Instead of forcing the entire screen to use a single palette (and thus a single 16 color subset of the 4096 available hues), the engineers built not one, but 16 palettes into the system, and made it so each scanline (horizontal row of pixels) could independently pick and choose from any these 16 palettes. Each scanline has a "Scanline Control Byte", or SCB, associated with it. The SCB controls which of the 16 palettes that particular scanline will use. (It also determines whether the scanline uses 320 or 640 mode; the upshot of this is that the screen can simultaneously display scanlines in both display modes!) This means that while individual rows are still limited to 16 colors, there can be up to 256 different colors displayed on the screen at once. This is the "256-color" mode you may have heard about.

Normally, scanlines are restricted to choosing between one of the 16 system palettes. However, it is possible to effectively assign each of the 200 scanlines its own unique palette, thereby making it possible to display 3200 different colors at once. Basically this feat is accomplished by keeping track of the location of the scanning beam as it travels down the monitor. Just before each scanline is drawn, the required RGB values for it are copied into the system palette. As you can imagine, this juggling act requires perfect timing. It also requires that interrupts be disabled — meaning many normal functions like disk access and even displaying the normal cursor have to be foregone. And because the screen is refreshed 30 times a second, the computer is kept very busy. Indeed, displaying a 3200 graphic leaves the CPU with very little muscle left over to do any "real" work. For these reasons, "3200 color mode" has largely been regarded as an interesting but useless curiosity.

If you've never known that the Apple IIgs is capable of multi-palette tricks you're excused, because few developers have seen fit to take advantage of them. One of the few hints that the IIgs is more colorful than it usually lets on is the ubiquitous rainbow-colored Apple icon in the menu bar of desktop applications. The fruit owes its stripes to multiple palettes. And if you've ever wondered how the palette editor in your fave paint program displays more than 16 colors, now you know. (Look closely and you'll find that each scanline has 16 colors or less, though.) Otherwise, developers haven't made much use of this nifty feature. Until lately, that is.

In recent months several major applications have come down the pike boasting support for 256- and even <gasp!> 3200-color modes.

DreamGrafix. DreamGrafix, from DreamWorld Software, is the first commercial paint program to fully support 640, 320, 256-color, and yep, 3200-color modes. Compared to its major rival (Beagle Bros.' *Platinum Paint*), *DreamGrafix*'s paint tools for 320 and 640 mode are nothing fancy. But for creating 256 and 3200 graphics from scratch, *DreamGrafix* is the only game in town.

From the moment you launch *DreamGrafix*, it's apparent that this isn't an ordinary Apple IIgs paint program. The menu bar across the top of the screen is familiar enough, but the little 3D style icons plastered along the lower quarter of the screen look a little like they might be the controls to a spaceship. And those little buttons don't just highlight when you click them — they animate, appearing to actually depress. There are lots of similar touches throughout the program, making it friendly and fun, though grumps might find it a touch unbusinesslike.

You choose which Apple IIgs graphics mode you wish to work in from the Edit menu. *DreamGrafix* gives you five choices: 640 mode, 16 Color, 256 Color, 3200 Color, and 3200 SBC. *DreamGrafix* allows you to freely switch between modes. Though the IIgs can do its multiple-palette tricks in 640 as well as 320 mode, *DreamGrafix* does not support this. In fact, *DreamGrafix*'s support of 640 mode is cursory at best; you can't edit individual pixels (only dithered "dots"), the paint tools don't take advantage of the higher resolution, and so on. For 640 mode painting, *DreamGrafix* is not a very powerful tool.

In (320) 16 Color mode, *DreamGrafix* is a little more a home, though many of the features we've come to expect from Apple IIgs paint programs since the days of *DeluxePaint II* are missing. In particular, *DreamGrafix* lacks facilities for creating masks and all but the most rudimentary of user-defined brushes. Also, some people may be disappointed to discover that *DreamGrafix* is limited to screen-size documents. Limiting 256 and 3200 Color documents to the size of the screen is understandable as these modes are intended purely for screen display, but this limitation doesn't make sense for plain vanilla 640 and 320 modes.

I suspect these shortcomings won't matter to most users, though, because they will already have at least one of the major paint packages (*DeluxePaint II*, *PaintWorks Gold*, *Platinum Paint*) and will use *DreamGrafix* primarily to make images created with them more colorful. This is where *DreamGrafix*'s other modes come in.

256 mode support. DreamGrafix's implementation of 256 color mode is straightforward. The only difference between painting in this and the 16 color modes is now you have 16 palettes to play with. *DreamGrafix*'s color editor is set up to allow you to modify the colors in three palettes at a time. To assign palettes to individual scanlines, you enter the palette editor. Here the program makes clever use of the IIgs's border colors to indicate which palettes are assigned where. This facet of creating 256 color graphics is a breeze.

Making effective use of 256 mode requires a fair amount of planning, and watching the color you are painting with change as you move up and down on the electronic canvas is a little disconcerting at first. Moreover certain types of images lend themselves better to 256 than others. For instance, a landscape with a sky that gradually lightens as it nears the horizon would be well suited to this mode. Other horizontally oriented images such as title screens and bar graphs are also easy to make more colorful in 256.

3200 mode support. The complementary 3200 Color and 3200 SBC are the sexiest modes DreamGrafix has to offer. Both are used to create the same kinds of 3200 color images; the difference lies in the interface. In 3200 Color mode, only the seven or so scanlines directly above and below the cursor are represented in true color. The rest of the image is depicted in grayscale. 3200 SBC stands for "Slow But Cool". In this mode, the entire working area is shown in true color, at the cost of some sluggishness on the part of the painting tools. (On my Zipped IIgs I found the performance to be satisfactory in either mode.)

In light of the fact that many people thought it was all an Apple IIgs could do just to display 3200 colors at once, the 3200 modes are a spectacular

technical achievement. However, as an artistic tool things are little more dubious. If 256 mode takes some planning, then 3200 mode seems to require the discipline of a nuclear physicist. For one thing, you are pretty much on your own when it comes to setting up the <ugh> 200 palettes. To ease the burden, *DreamGrafix* provides four drawing modes while working in 3200. In "Normal" mode, the color your drawing tool puts down is dictated by its position in the palette. Thus, if you select color two, the pencil will draw in the second color in the palette of each scanline you trace over. Depending on how your palettes are set up, the result may be a different color in each scanline.

The second drawing mode is called "Closest Color". In this mode, *DreamGrafix* tries to match the color you select in each scanline you paint in, regardless of its position in the palette. For instance, imagine you have a bright blue hue in position 3 of scanline 32's palette, and a similar color in position 5 of the palette of scanline 33. If you select bright blue to draw with, *DreamGrafix* will draw with color 3 when you are in scanline 32, and color 5 when in scanline 33.

The next mode is "Force Color". In Force Color mode, *DreamGrafix* always paints with the exact color you choose by modifying the palettes of each scanline you draw in. For example, let's say the third color in the palette of scanline 32 is still bright blue. If you select this color to draw with, bright blue will replace the third color in the palette of each scanline you draw in. This can cause pixels you did not intend to modify to change colors, so "Force Color" mode is most useful during the early stages of creating an image.

"Force Closest Color" mode behaves like a cross between "Force Color" and "Closest Color" modes. *DreamGrafix* will first find the closest match to the color you selected in the current scanline's palette, replace it with the selected color, and draw with that. This method can also produce some unwanted color changes, but in general is more controllable than plain "Force Color", and therefore more useful during the final stages of a painting.

DreamGrafix provides some other tools for dealing with the special problems of 3200 graphics — notably means for sorting the palettes and a special 3200 mode Fatbits editor — but make no mistake, creating images that take advantage of 3200 color mode with *DreamGrafix* is a tricky business that requires a lot of patience. The sample 3200 pics that are provided with the *DreamGrafix* package are telling. Not a single one was made from scratch with *DreamGrafix*; instead, they are all digitized images created by other means and converted to 3200 GS format! (DreamWorld deserves 40 whacks with a wet twinkie on a related score. An insert in the manual (correctly) deplores Apple II software piracy, while many of the 3200 samples are obviously copyrighted images belonging to other companies and are used without credit. Tsk Tsk.)

Well, if the guys at *DreamWorld* didn't create their own 3200 mode pictures from scratch, you don't have to either. Provided you have access to images in Graphic Interchange Format (or "GIF"), you can convert them for use on the GS in 3200 mode with *GIF 3200*, a shareware utility by Jonah Stich. GIF images are computer independent and may be any size with virtually any number of colors. Many GIF images will convert with near photographic quality with *GIF 3200*; the conversion can then be loaded into *DreamGrafix* for tweaking.

To convert existing images to 256 color mode, you can use *SuperConvert 3.01* from Seven Hills Software. *SuperConvert* is the commercial incarnation of Jason Harper's shareware classic *SHRConvert*. *SuperConvert* will load dozens of alien file types (including GIFs) and convert and save them back in most common GS file formats — including those that support 256 colors. (Unfortunately, *SuperConvert* won't convert files to 3200 mode.)

Another option is *The Graphic Exchange* from Roger Wagner Publishing. With the optional *Graphic Exchange Library Disk #1*, this program, which specializes in converting graphics among various Apple II formats (including 3200 color), can also handle seven types of GIF files and the PCX format commonly used by MS-DOS paint programs.

So now that you have the tools to create multi-palette images on

the Apple IIGs, what are they good for? Well, you could always print them out on your ImageWriter II and stick 'em on the fridge. Happily, *HyperStudio 3.1* offers a more interesting option. *HyperStudio* now supports 256 color images as card backgrounds without any fuss. In fact, *HyperStudio* will load mixed 320/640 graphics as well, though at the moment there isn't any practical way to create them. And there is at least one XCMD floating around out there that'll bestow *HyperStudio* with the ability to display 3200 mode pictures.

Just when you thought you'd seen about everything the IIGs could do with color graphics, a new crop of software comes along to push the barriers further out. And the nicest thing is that we already have the hardware to take advantage of the current Apple IIGs color renaissance.

Miscellanea

Roger Wagner Publishing is now shipping HyperStudio version 3.1. The esteemed Mr. Wagner encourages everyone to make the current upgrade. I quote, "The differences between 2.1 and 3.1 are incredible. 3.1 vs. 3.01 fixes a number of bugs, some fairly serious, adds some new features in the way of two new Extras, and generally uses memory more efficiently than 3.01 or 2.1." Upgrades from 3.0 to 3.1 are free if you send in your old disks or \$10.00 if you want to phone in your order. For information on upgrades from previous versions see the April 1992 issue of *A2-Central*, Miscellanea, p.8.22a. You can reach RWP at 1050 Pioneer Way, Suite P, El Cajon, Calif. 92020 or phone 619-442-0522.

One of our readers, J. Edward Jackson, informed us of a company that sells some very interesting pieces of Apple II hardware. Most of their offerings seem to be in the science/educational field. They carry an Air Track with Ultrasonic Measurement System, Air Track with Computer-Photogate Timing System, an Ultrasonic Measurement System, Computer Photogate Pars, Precision Timers, a Computer-based Radioactivity Set, and a

Geiger Tube-Computer Interface. I expect that if Art Coughlin doesn't already have their catalog, he will be the first to call them at 800-233-2490.

Am I the last to realize that Applied Engineering 800 number reads 55 4 MACS?

The Byte Works, Inc. has announced the release of ORCA/Debugger for programmers who use C and Pascal on the Apple IIGs. This is the only source debugger available for debugging CDevs, XCMDs, 320-mode programs and programs that run from the Finder. As an init, it can break into almost any C or Pascal program. The only limitation for use is with an interrupt handler.

The ORCA/Debugger has its own private text screen, so graphics and desktop programs can be debugged without interfering with the desktop display. It's designed to work with GSbug — they can both be installed at the same time. List price is \$50.00 and includes a 60 page manual. For more information contact The Byte Works Inc., 4700 Irving Blvd. NW, Suite 207, Albuquerque, NM 87114, phone 505-898-8183.

More new Apple II software. Bright Software is an independent group of programmers with a recent release. *GSymbolix* is an advanced scientific math program for the Apple IIGs. It integrates, solves, plots (2D and #D) formulas and also collects, simplifies, and expands them symbolically. It's programmed in assembly language. Bright Software can be contacted at P.O. Box 18, 4153 Reinach 2, Switzerland. On Internet: gudad@avalon.physik.unizh.ch or GEnie: A.Horstmann.

Broderbund has recently released two new programs on the Apple II platform. *The Treehouse*, a follow-up to their award-winning program, *The Playroom* is aimed at six-ten year olds. *Where in America's Past is Carmen Sandiego* is the fifth in the series of Carmen adventures. It comes with a 1,300 page encyclopedia that helps children to learn history as they track down the elusive Carmen Sandiego.



Ask (or tell) Uncle DOS

Corrections and amplifications

A few comments on "Smooth print tools" in the June issue...

The use of *Pointless*-generated bitmapped fonts with the LaserWriter was mentioned. What was overlooked is that the LaserWriter driver doesn't adjust bitmaps to its resolution and the print quality will still be unsatisfactory. For example, if you generate Monaco.48 it will be saved using the screen font resolution of 72 dpi. When you print this on the LaserWriter, it will be scaled up for the correct size at 300dpi. That is, the printer will print roughly 72 dots per inch at its 300 dpi resolution. The result isn't going to be pretty.

Since the article, Seven Hills has found that *Express* may not be particularly effective when used in conjunction with some printers and parallel cards due to the relatively high throughput of the parallel card; sometimes

even taking longer than direct printing. Since some of the overhead is involved in putting the spool file on disk, they're working on a solution to avoid the conflict by optionally spooling data to memory rather than to a disk file.

Express won't work with printer drivers that don't use a "printer port" driver such as the PRINTER and MODEM port drivers in the IIGs system software or the port drivers supplied with Independence or Harmonie. We recognized this would include the network printers (which use AppleTalk), specifically the LaserWriter driver. But we overlooked that the System 6.0 StyleWriter driver, which bypasses the port driver to operate at a higher baud rate, also would not be supported. Seven Hills was looking for a way to support network printers, but a printer driver like the StyleWriter's that sends data directly to the port hardware can't be redirected.

Also, in the first letter, "Basically a problem," last month we should have made a note that it wasn't a surprise that the patch needed to be removed from the reader's program.

The CHAIN bug "fix" pokes a byte that's in the middle of the BASIC.System interpreter (41859 decimal is \$A383); ProDOS 8 and BASIC.System internals are not guaranteed to be static from version to version, only the documented locations in the respective global pages, so indiscriminately poking in values meant for one version into another is dangerous. This POKE apparently ended up changing another routine that resulted in the error mes-

sage. You should PEEK the system interpreter version in location 49149 (\$BFFD in the ProDOS global page) from your Applesoft program to see if the value is "1" (BASIC.System 1.1 or 1.2, which had only minor changes) before applying the version-specific patch. (If you need to know the patch details, check Jerry Kindall's letter "CHAIN, STORE bug fix", p. 4.16, March 1988, with a cosmetic fix in "R fault", p. 4.29 of the May 1988 issue.)

It had been announced that the CHAIN bug had previous been fixed in BASIC.System 1.3 (see "Apple announces faster IIGs", June 1989). We work from the assumption that once Apple has fixed a bug they leave it fixed in later software (unless Murphy's Law exerts itself) so although the System 6.0 release notes don't specifically mention that the bug is also not present in BASIC.System 1.5 we would expect that it wasn't reintroduced unless testing shows differently.—DJD

One more correction before we get to the letters this month: A reader alerted us to the fact that the area code given for *Library Talk* magazine in the "Marketing Educational Software" article was incorrect. The correct area number is 614-432-7107.—edr

Just a (bad) memory

I am not sure if I have a virus or I am the victim of a nasty bug in Appleworks GS. I am also sending an account of my recent experiences with Glen Bredon as part of my order for Apple.Rx. I am hoping that either he or you

might be able to shed some light on this.

I own an Apple IIGS with 1.75 MB RAM and a Vulcan 20 MB Hard drive. I use *Prosel 16* (v. 8.44) as a front end. *Prosel* identifies the system software as \$0303. I have System 5.04 installed (from the updates on A2 on Disk). I use *AppleWorks*, *AppleWorks GS*, *Talk is Cheap*, and *ShrinkIt* often. *WordPerfect* is also installed on the hard disk. I have no stolen software; everything I use is registered.

The Vulcan is set up with two 10MB partitions called AE1 and AE2 respectively. AE1 is the boot partition and contains all my software. AE2 contains documents and other things I produce.

The following incident has now occurred twice. The first time was about a month ago. The second time was two days ago.

I was using *AppleWorks GS* page layout module for some fairly complex work. I save frequently as a habit and all appeared well. Just before I quit, I save, then exited, only to be greeted with a dialog box that read, "Please insert Volume AE1." The machine would not reboot and reported "Unable to Load ProDOS" when I tried. I then booted from a floppy and had to look at AE1 and AE2 with the Finder.

What a mess. AE1 had been renamed IE1 (AI1 the first time it happened last month) and every other folder and file had been renamed with what appeared to be random characters. I attempted a volume repair with *Prosel's* utilities. Even though *Prosel* has managed some pretty miraculous recoveries in the past, the volume repair utility reported that too much damage had been done to the directory blocks (block 2 in particular). The same was true of AE2. It looked, in fact, like junk had been written over both volumes. I had no disk in my floppy drive, so I can't say whether this would have been affected as well.

I had to reformat both partitions (clearly the partition info had not been damaged), reinstall the software and restore my work from the last backup disk. I did not lose much except for time and temper. In my view, I was prepared to accept (reluctantly) the first incident as an unfortunate glitch, perhaps due to a power surge or brownout while I was saving (or something). The same thing twice is more sinister and means either a virus or a serious and nasty bug in *AppleWorks GS* or the system software perhaps. If it is a virus, I have no idea where it came from. There are only two possible sources—*WordPerfect* (I received and installed an upgrade recently from *WordPerfect Pacific*) and **A2-Central** on disk (a source of much enjoyment, shareware and public domain goodies). Neither possible source is very likely.

I have written to Glen Bredon to obtain a registered copy of *Apple.Rx* which I intend to install. I have also asked for his comments which I will gladly pass on to you. I would be interested in any information about this episode. Perhaps others have also been affected.

Ross Barrell
Warwick, Queensland

Starting with the good news: viruses on the Apple II are rare. In general the only strains we've confirmed are the two ProDOS 8 viruses (*CyberAIDS* and *Festering Hate*) which are hardly "stealthy" since they bloat files and access all drives at startup (unusual for a ProDOS 8 program), and a series of pesky but otherwise innocuous boot-block viruses (like *Load Runner*) that attack the IIGS. We don't recommend against caution and purchasing and using a virus detector is a reasonable precaution, but don't lose sleep over the possibility a virus may strike. We don't; we keep checking the disks we use (including those we send out) but we've never been hit with an Apple II virus and only once with a Mac strain when viruses were new news.

The bad news is that a computer is a complicated piece of machinery and there are lots of other things that can go wrong. We are so impressed when they work correctly that we forget how fragile they can be.

So the first thing we'd suggest is that when something goes "wrong", don't lock in too soon on the idea that you've been hit with a virus. It's very likely something else. That shouldn't necessarily be comforting to you; as system problems go, viruses are relatively easy to detect and cure because their behavior is usually well-defined.

We work with all the monthly disks we send out as well as the programs you mention with the exception of **WordPerfect GS**. We've not had any problems we can attribute to severe bugs in the programs (we do keep our software updated to current versions, which is probably a good precaution since earlier versions may not be tested with current system software).

Because the IIGS system software is large and complicated one of the most common problems is damage to some part of a program in memory. This can happen for several reasons and will result in a multitude of problems. If part of a disk buffer or driver is damaged before writing to disk the damage can be catastrophic: worse even than a typical virus. Most people are overly concerned with viruses and not respectful of just how fragile the computer environment is. When something goes wrong it may be a normal failure of the hardware or software; there are some general things you can try to track the problem down, but the final analysis often requires closer examination by someone with specific expertise. Your dealer may or may not employ someone qualified.

The first step is to try and confirm that the computer hardware is not at fault. To try and confirm this, run a basic memory test. We prefer *Claris's MemoryTest.CL* included with *AppleWorks GS* because its refresh test mode checks the stability of static data in memory over time (albeit only a few minutes). If it checks out, try letting it run overnight. If it still checks out, start suspecting a software conflict instead.

If the problem does seem to be hardware, you should remove "extra" cards (those not required to start up) until the system starts

behaving reliably. If the system stabilizes, make a note of the cards not in the system and then add them back in one at a time. If a specific card seems to correlate with problems then report the minimal configuration that causes the problem to the manufacturer of the card to see if they know of a conflict. If the problem seems to crop up with various combinations of cards then start suspecting that your system's power supply may be overloaded (cooling or a more robust supply may help).

If the problem seems to be software, then you should verify that the boot volume's file structure is intact. You can use the Finder's "Validate" command under the "File" menu to check files and the "Verify" command under the "Disk" menu to check disks. If it's okay and you feel your last backup is current enough, proceed with the next step. If it's not okay, backup what you need to (and can). If it's not okay, and you haven't made a backup in recent memory, and you've lost an important file, you may now accept that you've been properly educated.

After you've saved what there is, you're safe to test. Fix or restore the boot volume if necessary so you can try the same things you were doing when things went wrong. The next idea is to boot with a minimal system; either from floppy (make a copy of the basic /System.Disk and add only essential drivers) or by reducing the hard disk system to essentials (the easiest solution is probably to rename the "System" and "Icons" folder to something like "System.old" and "Icons.old" and installing a new minimal system). If the problems go away the problem is probably one of the "extras" you're using; you can try to ferret it out by adding the extras to the System folder a few at a time and watching for problems, then removing any suspects one at a time until you locate the likely culprit. Yes, this is time consuming; the rule of thumb is to try and keep your System folder as "minimalist" as possible. (One may normally assume the Apple system software is well-behaved, so concentrate on testing the non-Apple additions unless you have a good reason to suspect one of Apple's.)

Because these system software extensions (drivers, setup files, desk accessories, and CDevs) are such a frequent problem many of the non-essential ones can be disabled when booting System 6.0 by holding down the "shift" key as you restart. The (graphics) title page will notify you that "Inits/DAs are disabled" (also CDevs, but not drivers). If the system booted in this way is more stable than after a normal boot then start suspecting that you've added something that may be causing problems.

This is not a sure-fire scheme, but it will often locate the most likely problems. It also isn't extremely detailed, but then I don't know any repairmen who can fix your television or car via the phone or mail.—DJJ

Talking about Apple

Your May issue on AppleShare connectivity couldn't have come at a better time, as I was in the middle of setting up a network in my home office, connecting two Apple IIGs computers with a Mac LC.

I finally discovered the Mac is good for something after all: as a peripheral to my Apple II's.

Here I am writing this letter on my Apple IIGs (one of two I own) in the dining room. I wanted a computer in the living area of my house so I would be around the kids more, but it had to be a setup with minimal hardware so as to be unobtrusive. There is no hard drive on the IIGs, but it can access 160 megs of programs and data directly from my Mac stored in a dark corner of my basement (where the Mac belongs). An AppleTalk ImageWriter is also in the basement. It takes a little getting used to seeing no disk access for most activities—and I still sometimes check if files really were saved.

With System 6 on my Apple IIGs and System 7 on my Mac, I have found these two computers work together better than ever. I can place Mac or ProDOS disks in the drive of either computer and read them. I prepare documents in AppleWorks on my IIGs, save directly to my Mac, then read them directly with *ClarisWorks*. Or I may prepare a file with *ClarisWorks*, and save it as an AppleWorks file and load it directly with my IIGs. Now when someone asks me "Do you want the file in Apple II or Mac format?" I just say "yawn—doesn't matter—it's all the same anyway." Of course, my Mac LC has the IIe card in it—making it really useful.

A network is good in a business environment, but one may ask "what use is it to me in my home?" I discovered a very valuable home use: My kids play computer games. I know enough to give them warning before dinner to let them finish up. At dinner time, I kept calling but always heard the response "just a minute" over and over. Frustrating. No more, though. Ten minutes before dinner, I go to the file server and set it to disconnect in 10 minutes. They get a warning dialogue box, and 10 minutes later they magically appear at the dinner table. <grin, snicker>. (Does this story qualify me for a prize as a "clever or creative letter" as noted in the June issue?).

In setting up AppleTalk, I moved the SCSI card in my home office Apple IIGs (ROM 01) from slot 7 to slot 2, and changed the control panel setting to "Your Card." A friend told me I could still use *ProTerm* and the modem plugged into the modem port. I didn't believe it, but it worked! Do you know why?

After installing the network, I experimented with setting up many "devices" (partitions on hard drives, RAMdisks, and floppies) on my IIGs and at one point had 16 devices on line. I discovered AppleWorks crashes on startup if more than 12 devices are on line. I was told that ProDOS 8 will only recognize up to 14 devices — 2 for each of the slots from 1 to 7. "They" were wrong - as usual! I was able to access all 16 devices from within AppleSoft BASIC. True, only 14 could be found by using the PREFIX Sn,Dn.

But if I specified PREFIX /pathname all 16 devices could be accessed. When I entered the ProDOS command CATALOG of Mac drives on the network, invalid ProDOS names showed? where the characters were invalid. I've learned to name disks/folders/files on my Mac with valid ProDOS names.

Although I understand there is a special edition of AppleWorks for networks, I found my standard edition v3.0 worked fine, except if you tried to save to the same hard drive at the same time from both Apple IIGs's. The AppleWorks that tried to start the save second got an "error trying to save" message. When one computer was saving a large file, I could watch the save in progress (as the TEMP.000000 file), and increasing in size, from within AppleWorks on the other computer. I could load the TEMP file from one computer while it was being saved by the other.

I teach "Introduction to the Macintosh" (I know, I'm a traitor!) at a local university night school. In a recent class, a student said she was taking the course because "I just bought an Apple IIGs and this course is the closest I can find on something that works like it." Needless to say, I immediately modified my course on the fly to "sneak in" some IIGs specific stuff. I also gave her some additional pointers during breaks and after class. (Hope you also gave her our address and phone number, Jim! <grin>)

In my preparation for each Mac course, I have to mass produce a couple of disks for the students. Naturally, I do in on the Apple IIGs which can be so much faster at disk copying. And to recycle disks my Mac students need to learn about formatting, I use the IIGs Finder "Erase Disk" command—which doesn't go through a long format procedure if the disk is already in a recognizable format. Ever try to "erase" a disk from the Mac Finder? Sighhhh....

Jim Low
Toronto, Canada

Your trick to get the kids to dinner sure deserves something, Jim. This tip alone may convince my husband that we need a second computer so that we can network. Tell you what, if it works.....—edr

People helping people

Reference the June issue of **A2-Central**, "Don't interrupt," p. 8.31, from Robert Burger of France. The problem stated was a sound interrupt error after installing a Transwarp GS.

I had a similar error after installing the 32k upgrade to my TWGS card. My Apple IIGs would crash at random times for random reasons, including the same sound interrupt error. The problem cleared up when I installed a new ribbon connector on the TWGS board. The ribbon connector is (was) a free replacement from Applied Engineering. A \$5.00 shipping and handling fee was required. According to AE, some ribbons would fault due to thermal build up. My bad ribbon was identified as having a gray plastic connector on the TWGS side. The replacement ribbon has black connectors at both ends and a black dot on the ribbon itself. Since

installation of the new ribbon, all such crashes have been eliminated.

Jim Werner
Palmer Lake, Colo.

In your June issue, Robert Burger from France wonders about how to cure Sound Interrupt errors. You mention, as one possibility, cleaning the Ensoniq chip. Another possibility may be worth considering: some NDA's cause the problem. In particular, the Beagle Desk Accessories *Phone Filer* causes an Unclaimed Sound Interrupt error when accessing a P8 file after using it. (For some reason, this does Not happen when accessing it from *Wings*.)

Kirk Hollingsworth
New York, N.Y.

I had the same problem as Robert Burger for awhile. I came to the conclusion tht it was the beast's erratic way to react to overheating (mine is packed with cards). I took off the cover, and since then it behaves. I can't guarantee it's a universal cure, but in this case, it worked.

Chris Marker
Paris, France

Memory clarification

I'm writing to clarify a previous letter concerning printer problems. Seven Hills has determined that the problem stems from a memory shortage. Apparently System 6.0 takes a larger chunk of memory than System 5.04. A new beta version of the *Independence Deskjet* 500 driver corrects the problem by breaking the output into more bands, thus requiring less operating memory. Although one might expect that this would slow printing, it doesn't. In fact, in some cases, documents actually print faster.

Another partial solution is to use the *Memo-ry Bar NDA* by Dave Lyons to compact and purge memory before printing. The purge feature is undocumented. To execute a memory purge, hold down the open-apple key and click the Compact button.

I can't say for sure but I would guess that the problems that other people have been experiencing might also stem from memory shortages.

Curt Clifton
Ames, Iowa

It's not alarming

Although Fred Greatorex did not state exactly which *AlarmClock.NDA* he had been using in his letter (It's Alarming, June 1992), I did some checking and *Alarm Clock NDA* (v2.1 by Bill Tudor) works just fine with both v1.0 and v1.1 of HyperCard IIGs. *Alarm Clock NDA* even knows how to stop displaying the menu bar clock when *Hypercard* hides the menu bar. No one has reported any problems with *Alarm Clock NDA* and **any** IIGs desktop application.

After reading the letter again I think I have the answer. As is clearly stated in the documentation, *Alarm Clock NDA* **requires** System Sof

ware 6.0 or later. I suspect that your reader was trying to use the NDA with an earlier version of the IIGS system software.

Bill Tudor
GEnie: W.TUDOR

Peek publishing

I have been writing programs in Applesoft since the early 1980's, initially for the Apple II Plus and currently for the Apple IIc. However, I have never had the benefit of any "Peek/Poke" charts or lists other than those issued with early Beagle Bros programs. These give certain essentials, but they do not include the more technical addresses, e.g. Peek(978), which tells you whether DOS 3.3 or ProDOS is installed (**A2-Central** v.1, No. 0, p. 5a). I know there are many other such Peek addresses (like one to tell you whether the computer is a II Plus, IIc, IIe or IIGS, one that reveals how much memory is installed, etc.) but where can one find these things out? I am sure that they are listed in some technical reference, but which one?

Leonard Fromm, Jr.
Cold Spring, Minn.

*Over the years there have been several books that have had fairly complete peek/poke charts, but they're out of print and out of date. What I'd recommend is the program **BASIC Help**. Just type "HELP" at the Applesoft prompt and up pops an online manual for Applesoft and ProDOS. The program includes information on command syntax, math and logical operations, error messages, and peek, poke, and call locations. **BASIC Help** is listed in our catalog (IS-002, \$29.95). —TW*

AE Fax Problems

It is widely known that the new System Software for the Apple IIGS and the AE Fax software do not work together. The real pain is caused by the FaxCDev. Making this CDev inactive eliminates the problems. But you no longer have use of your fax!

Here is a solution—Make sure your CDev is inactive so you do not have any problems. Choose as the Direct Printer the FaxDrive via the modem port. Print the file the normal way (Command-P) but instead of saying "Now" let the file be put in the queue to print at a later time. Now quit the application you are using. Choose the Fax Log NDA under the Apple menu in the Finder where you'll see that the fax to be sent is in the queue. Make this file active and reschedule it to time: Now. The fax will be sent right away without any problems.

The disadvantage is that you don't have the opportunity to use the Phonebook and the fax will be sent using the parameters (background or not, etc.) set by the FaxCDev the last time you used it. Of course, this is better than not having use of the fax at all. I do hope that there will be an upgrade soon, one that is compatible with System 6 and that will include the use of a customized front page.

Peter van Dongen
The Netherlands

AWGS-Printer Help needed

I recently bought a second-hand Apple IIGS and I'm very pleased with it.

Printing from AppleWorks GS is very slow but that's all in the game. However, there was a little problem that I solved myself. Everytime I printed from AppleWorks GS to my Epson RX-80 printer, from the second page on I got the three letters K N H, at the top of each page, printed as text, not in the graphics mode the rest was printed in. I use a Cirtech parallel card interface card. So I decided to load the parallel card driver in memory (load/system/drivers/parallel.card,A\$2000,T\$BB) and use the monitor to see what was there.

The three annoying letters only appeared twice in the locations \$211e, \$211c and \$2120 and \$2139, \$213b and \$213d.

I changed all 6 to AO, <space> and bsaved /system.disk/system/drivers/parallel.card,A\$2000,L1635,T\$BB.

This solved the problem, and everything else worked just fine. What struck me was that the code of the driver seemed to contain a lot of useless code.

Having done this successfully, I went on and loaded the Epson driver, to see if I could get some more speed out of my printer. After loading, (load/system.disk/system/drivers/epson,a\$2000,T\$BB) I discovered the graphics commands (esc * n) at 3d7c, and changed "n" from 04, for CRT graphics (whatever that is), to 02 for double density, which makes the printhead move twice as fast according to the printer manual. After bsave/system.disk/system/drivers/epson,A\$2000,L23136,T\$BB, the only problem is that the printer prints a little less wide. When changing the code to 00, for single density, the output becomes too wide for the paper.

Additionally, the printer prints every line 4 times, even when there's nothing to print at all. I think that one time would be good enough for me. Does anybody know how to change that?

Eduard de Kam
Amsterdam, The Netherlands

More printer help

I would like to use a Star Micronics LV1215 parallel printer with AppleWorks 3.0 to get long database printouts on the 15" carriage. Can you suggest an interface that would allow my Apple IIGS and AppleWorks 3.0 to talk to this printer? I was told that the Star Micronics emulates the IBM ProPrinter.

Bryan Johnson
Lake Wales, Fla.

Most common parallel cards should work fine; my suggestion would be to pick one of the currently popular ones like the Orange Micro Grappler Pro or the Applied Engineering Parallel Pro. I have used an original (1980 design vintage) Apple Parallel Card "clone" with good results. (And with excellent results using Harmonie and Pointless with AppleWorks GS and a Citizen 145 printer.)

I don't see a ProPrinter emulation listed in AppleWorks 3.0's printer installation menu. If

the Star emulates an Epson series printer one of the Epson model setups (MX, FX, or RX series) should work. If not, you may have to define "custom" printer (check through the files on your original AppleWorks disk; there are database files with codes for various printers and interfaces).

To use "wide" printers, you'll need to set the interface card to print longer lines. AppleWorks sometimes defaults to using "Control-I 80N" for the "Interface Cards" setting for a printer; that tells the printer interface to execute a carriage return after 80 characters with line linger that 80 characters. You'll probably want to change this to something like "Control-I 132N" (132 characters per line).

Finally, the word processor seems to have a limit of a platen width of 13.2 inches. You may find that you can't use quite all of the printer's width even after adjusting the settings.—DJD

HyperQuestion

Is there now a way to use Macintosh stacks, in one way or another?

Hans De Vries
Oegstgeest, The Netherlands

*There is a relatively easy way to convert Macintosh Hypercard stacks to the Apple IIGS. **HyperMover** (DA-030 \$15) is a utility from Apple Computer, Inc. designed to do that very thing. You do need both a IIGS and a Mac to accomplish the deed however. **HyperMover** comes on two disks, one for the Macintosh and one for the Apple IIGS. Basically, all you do on the Mac side is run the program, open the stack you want to dismantle, and let the machine do its thing. Once dismantled, you run **HyperMoverGS** on your favorite computer and reassemble the files that were created on the Mac. You can also run the program the other way to convert IIGS stacks to the Mac.*

*A couple of important points you need to be aware of are that the program cannot convert XCommands and that the differences in screen resolution and color between the two machines usually mean you have to redo the graphics. **HyperMover** handles the most tedious parts of the transfer, however.—edr*

Simply Mahvalous

I just wanted to thank you for your fine article in the April 1992 issue entitled, *Programming is Simple*. The timing was perfect for me since I was right in the middle of working on a stack that needed a little something extra to spark it up. After reading the article and applying the techniques to my stack, I can safely say that *SimpleScript* was the missing spark. I thoroughly enjoyed Jeff Smith's non-threatening, non-technical style of writing. Granted, I have looked at the manuals that came with *HyperStudio 3.0* but I never had the desire to read them. The detail and patience presented in Jeff's article inspired me to action (sort of like an NBA-New Bill Action). He is right! Programming is Simple and now I feel like an expert programmer <snicker, snicker>

I encourage more articles like this which attempt to teach us users tricks and short cuts of a particular program. He mentioned two shortcuts alone that made the article worth reading: 1) Command-Shift-Tab get you in and out of the Edit (or button or text or graphics) mode; 2) Holding down the Command key while clicking on a button with *SimpleScript* gives you a great Debugging window. I'm not sure I would have found that last goodie in the manual. If you missed this article and have felt programming to be for "others", give this article a second look.

Bill Cross
Arroyo Grande, Calif.

Marketing article update

Many states these days seem to be doing a better job of setting up a central location to evaluate and preview new educational software. Some of these 'clearinghouses' and 'technology centers' allow teachers to come and preview hundreds of different titles of educational software.

And, interestingly, some of these organizations are making an effort to reach out and support small publishers as well. One notable organization, the Indiana Clearinghouse for Educational Technology, even offers to send out mailing labels of computer-using teachers in the state. They also display catalogs and flyers at

their Preview Center.

For further information, contact: Benjamin Crosson, Preview Center Manager, Indiana Clearinghouse for Educational Technology, 620 Union Drive, Union Building 123, Indianapolis, IN 46202. (317) 274-8001, (800) 222-4223.

Hopefully Indiana's bold initiatives will serve as model for other states to follow."

Phil Shapiro
Washington, D.C.

Fax alot

With your help, I was able to print a fax with my LaserWriter from within AppleWorks 3.0 on an Apple IIgs (ROM 01, two 3.5 drives, no hard disk, an ImageWriter connected directly into the IIgs, the LaserWriter is on an AppleTalk network shared with several Macs, no server). As you remember, the main problem was to have enough space in an 800k disk for a bootable system disk with the driver for the LaserWriter.

Here is a summary of what I have done with your help. Perhaps it would be useful to somebody else. In addition, I have found a bug on the emulator program and I suggest an imperfect solution which may be of interest to those writing French text.

1. Initialize a blank 3.5 disk
2. Install "Appleshare on a 3.5 disk" using the Installer.
3. Remove "Appleshare" (not Appleshare on a 3.5 disk.)
4. Install "LaserWriter."
5. Install Appletalk on slot 7 (I have done this from the "old" control panel, i.e. access with oa-ctr-esc.)
6. Leave the printer port to slot 1.
7. Connect the ImageWriter cable on the printer port and the Appletalk cable on the modem port.
8. Reboot the System with the new System disk that has the LaserWriter installed.
9. Open AppleWorks 3.0 from the System disk.
10. Install a new printer, "ImageWriterLaser" in slot 7. In slot 1, leave the ImageWriter connected directly to the computer. I have changed "needs feed line after each return" to "yes" and interface card to none.

It works! I can switch between the ImageWriter in slot 1 and the "ImageWriterLaser" in slot 2 without problems. There is also no problem sharing the LaserWriter with the Macs on the network.

All the features like bold, underline, superscript, subscript, etc work fine. Output in German and Spanish is great. The only problem I have is that the codes for French do not work. According to the ImageWriter manual the sequence is:

ESCAPE Z CONTROL-A CONTROL-a` ESCAPE D CONTROL-F CONTROL-a`. Since this sequence works fine with the actual ImageWriter, it must be an error on the emulator.

I have found that you can get French output writing the following sequence:

ESCAPE Y CONTROL-A CONTROL@ ESCAPE D CONTROL-F CONTROL@ SPACE SPACE

However, you can use the special code only one time per document. If you use it a second

time, different errors will appear.

Luis Fontbote
Geneva, Switzerland

Bad database!

In the past three months, I have received two disks containing an AppleWorks data base file that could not be restored with Change-A-File. One was on a 5.25 disk and the other on a 3.5 disk. Each disk was almost completely full when the file was saved to disk. Each file was corrupted in exactly the same way:

1. In the directory entry describing the corrupted file, the block number of the index block of the file (relative entry bytes \$11-\$12) pointed to a false index block containing only a single block pointer.

2. The single block pointer pointed to the true index block of the file. The initial block pointer of the true index block had been changed to \$0000. The remainder of the block pointers were correct.

3. The correct initial data block of the file was the block just before the false index block (block number in directory entry bytes \$11-\$12 minus one.)

The file can be restored as follows:

1. Copy the original disk containing the corrupted file and save the original. Use the copy.
2. Using a zap utility, such as Prosel Block Warden, find the directory block containing the file entry describing the corrupted file. Find the relative bytes \$11-\$12 and obtain the pointer to the false index block. Save this pointer.
3. Read the false index block to obtain the block number of the true index block.
4. Change directory bytes \$11-\$12 to point to the true index block. Change the initial entry of the true index block to the saved pointer minus one.
5. Load the file into AppleWorks and re-save it to another disk.

I have not yet been able to reproduce the conditions that produce the corrupted file, but I would be wary of saving an ADB file to a nearly full disk. If any of your readers believe they have a file corrupted as noted above, I would appreciate it if they would send me a copy of the disk with the corrupted file. I am trying to determine the conditions that produce this problem. Please include return postage.

Harold Portnoy
1431 Woodward
Bloomfield Hills, MI 48302

More reader input

Your readers may be interested to know that CD driver software for the Apple IIgs is available from Trantor Systems Ltd., 5415 Randall Place, Fremont, Calif. 94538-3151, phone, 513-770-1400.

Paul Gorman
Portland, Ore.

A2-Central™

© Copyright 1992 by
Resource Central, Inc.

Most rights reserved. All programs published in *A2-Central* 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.

Publisher: **Tom Weishaar**
Editor: **Ellen Rosenberg**

with help from:
Denise Cameron **Dennis Doms** **Sally Dwyer**
Dean Esmay **Richard Ginter** **Jeff Neuer**
Jean Weishaar

A2-Central—titled *Open-Apple* through January, 1989—has been published monthly since January 1985. World-wide prices (in U.S. dollars; airmail delivery included at no additional charge): \$34 for 1 year; \$60 for 2 years; \$84 for 3 years. All back issues are currently available for \$2 each; bound, indexed editions of our first six volumes are \$14.95 each. Volumes end with the January issue; an index for the prior volume is included with the February issue.

The full text of each issue of *A2-Central* is available on 3.5 disks, along with a selection of the best new public domain and shareware files and programs, for \$90 a year (newsletter and disk combined). Single disks are \$10. Please send all correspondence to:

A2-Central
P.O. Box 11250
Overland Park, Kansas 66207 U.S.A.

A2-Central 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 *A2-Central* for distribution to others. The distribution fee is 20 cents per page per copy distributed.

WARRANTY AND LIMITATION OF LIABILITY. We warrant that most of the information in *A2-Central* is useful and correct, although drift 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. OUR LIABILITY FOR ERRORS AND OMISSIONS IS LIMITED TO THIS PUBLICATION'S PURCHASE PRICE. In no case shall our company or our contributors be liable for any incidental or consequential damages, nor for ANY damages in excess of the fees paid by a subscriber.

ISSN 0885-4017

Genie mail: A2-CENTRAL