

A journal and exchange of Apple II discoveries

Sculley committed to Apple II

John Sculley spoke to educators on Apple's future plans during the *Apple Vision 90's* event broadcast to 21 Midwest cities on April 24. Comments regarding the Apple II's role in this commitment were a significant portion of the speech.

Mr. Sculley made several introductory remarks about Apple's involvement with education, and commented on his perception that we are moving from the industrial age economy based on manufacturing from raw materials into a global economy based on the value of information. His thrust regarding Apple's partnership with education was a concern in insuring that our society did not become "resource poor" in the information age. Apple's recent focus on Macintosh at the higher education levels and the Mac's introduction into the kindergarten through grade 12 (K-12) curriculum was discussed. He then used this as the basis to address the challenge that Apple is facing ("Some of our competitors have said that we no longer care about the Apple II") by responding to the question "What is our commitment in the future?".

Mr. Sculley's remarks included several very positive statements:

"There is no computer in the world that has changed education more than the Apple II. It's a proven tool. Its one that you as educators helped shape. It is one that we are firmly committed to in the 1990's."

"...the way the Apple II is designed, it's very modular in concept. We are able to add technology to it..."

He used an interesting new term in explaining how multimedia creates a more effective learning environment.: "Video overlay means that the *Macintosh IIgs* becomes a true multimedia environment." (emphasis ours).

The most heartening news? "Another way to look at our commitment is to look at where we are spending our research dollars. I can tell you that in this fiscal year, 1990, that we have increased our research and development behind the Apple II by some 20 per cent". So we're not walking away from the Apple II at all, but we recognize that not every technology is right for every situation, that it is important that you have choices."

In regard to those choices, some of Apple's 1990-91 strategy was revealed along with comments regarding their current technology. The first strategy involves creating low-end Macintosh systems which retain their system software consistency with the current Macintosh line. Also, Apple will be trying to solidify compatibility bridges between the two product lines. This has already been achieved to a large extent for peripherals such as printers and disk drives. Data compatibility can be facilitated through the two computers common use of a 3.5 floppy format, and the connectivity provided by Apple-Share. Adding the Mac's ability to read MS-DOS disks, Mr. Sculley proclaimed that no other manufacturer can currently offer that flexibility in data exchange.

Mr. Sculley also revealed that Apple is working on an Apple II version of HyperCard, referring to the vast array of user applications ("stacks") generated by HyperCard's "end user" technology. HyperCard does not require a technical programmer to create applications, and Mr. Sculley asserted that Apple did want to bring that same capability to the Apple II as another way "the two products can leverage off each other in a mixed environment".

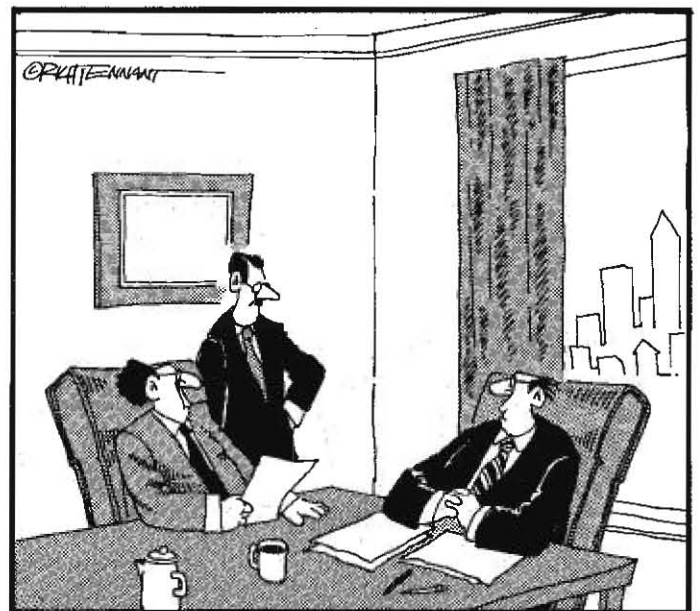
He stressed Apple's deep commitment to education, education reform, and the K-12 market. His concluding remarks included:

"We also have a very strong commitment to two technology families, the Apple II and the Macintosh. Each has their own set of strengths. We want you to be in a position to choose what you feel you need to be able to do..."

One other future product was mentioned: a no-compromise Apple II emulation card that will operate inside of a future Macintosh product.

Apple's Vice President in charge of Education, Dr. Bernard Gifford, spoke next in a session where the emphasis was on computer based educational tools for a "true learning society". He began with a discussion of Apple's current programs such as the *Apple Classroom of Tomorrow* project and Apple's grant programs. He also discussed *Geographic Television*, the multimedia American history project for the IIgs born out of the partnership of Apple's MultiMedia Lab, *National Geographic*, the state of California, and LucasFilm.

The first response during a question and answer period regarded the development of Macintosh educational software, and whether Apple II software was still being developed. Dr. Gifford revealed that a guide consisting of over 250 programs selected from a group of 1000 Mac educational packages would be available. He expected that the catalog might expand to 1000 application by next year. He also revealed that 900 new Apple II applications had arrived since the beginning of 1989 (over the course of the last year and a half), and that a new booklet would describe these programs. He added "...all of the applications included in this booklet, I think, clearly demonstrate that there is a heck of a lot of life left in the Apple II platform; it is still generating enormous creativity on the part of developers." He also



"... AND TELL THE SCHOOL BOARD THAT SYSTEM INTEGRATION SHOULDN'T BE A PROBLEM. THE NEW COMPUTERS ALL SEEM TO BE PRETTY MUCH THE SAME COLOR AS THE ONES WE HAVE."

mentioned that there is a tremendous increase in AppleShare-aware software.

The comments were received well by the attending audience in Kansas City. I did find a few questions unanswered there; notably, whether Apple will also direct its marketing and evangelism efforts more effectively in establishing a commitment to the product lines based on user concerns, rather than on product ego. Specifically in regard to the IIGs, it is Apple's handling of the Apple II, more than any technology gap between the Apple II and the Mac, that has resulted in user concern about the commitment and direction of the company. As if by illustration, at the Vision 90's event the only Apple Computer brochures in view were for Macintosh products.

But we later received a videotape titled *The Partnership Continues...* as part of the package which included the promised software catalogs (impressively, this arrived about a week after the *Vision 90's* event), and this tape referred repeatedly to the synergistic combination of IIGs and Macintosh systems side by side, and also to the longevity of the "older" Apple II systems which are still in use. John Sculley also asserted that an Apple II team had been brought together for more coordinated development. Will this be the face that Apple maintains? It seems rational; a survey in the October 1989 issue of *Electronic Learning* (p. 24) indicates (for those states with separate figures for Macintosh and IIGs systems) that there is a preponderance of IIGs systems to be supported.

I believe most Apple II users wish the Macintosh well. Apple apparently does realize that the "computer for the rest of us" is not as universally acceptable in its present form as Apple would like. On an American Broadcasting Corporation *Business World* broadcast, John Sculley attributed Apple's decision to reduce the price of the supposedly "hot" Mac Portable by \$1000 to the fact that most purchases seemed to be by current Mac enthusiasts. Since the product was not expanding the Mac market as they had hoped, a price reduction was implemented to see if that made the system more attractive to potential new customers.

The Apple II represents a significant gap in Apple's recent marketing coverage. The majority of Apple II users selected their systems after the Macintosh was released, and I think they also expect to have a reasonable upgrade path until they decide to move to another CPU. With System 5.0, Apple delivered on some of the technological promise of the IIGs; demonstrations of what may become elements of some future system software release (such as enhanced IIGs Sound Tools) indicate that promise can be fulfilled. The remaining question is whether Apple's marketing and evangelism elements will make the IIGs product viable so that Apple II customers can feel that they have been given a respectable return on their investment with Apple. Faster hardware and greater vertical resolution (to solve the 640x200 aspect ratio problem when going from display to printout) would also be nice.

I am a realist, and I don't think I'll believe all of what was said until it actually transpires. But the most important aspect is that many positive things were said about the future of the Apple II, and they came from the person who is in the best position to see that they are delivered on. I think the real key is in John Sculley's reference to the "Macintosh IIGs"; if that is an indication of how they will try to market the IIGs, I think they will lure some of their alienated customers back into the fold.—DJJD

Miscellanea

Attack of the killer SCSI cards. We received our *Apple II High Speed SCSI Card* the first week of April. The speed increase in normal use was not as blindingly obvious as we had initially hoped (about a 15-20 per cent reduction in GS/OS boot times, and about a 30 to 50 per cent reduction in file access times), but we were overjoyed to discover that the promised compatibility of the card and new drivers with a wider range of SCSI devices seemed to be accurate. We are now using a SyQuest removable-cartridge (44 megabyte) hard drive mechanism attached to an Apple card with the new drivers; a configuration that did not work properly with the previous Rev. C card and drivers.

For pure speed, nothing would appear to beat the new \$289 *RamFast* SCSI interface from C. V. Technologies, 9431 Saddlebrook Lane,

Suite #2C, Miamisburg, Ohio 45342, 513-435-5743. We've had one here briefly to test, and our IIGs boots as fast (or faster) from a hard disk attached to the *RamFast* as from a RAM disk (about 8 seconds for a minimal System 5.0.2 configuration).

The *RamFast* includes an on-board ROM disk with the configuration program and driver (you can't misplace the software unless you lose the card!), as well as a set of configuration switches. The switches allow you to set the default configuration for the interface's SCSI identification number, DMA compatibility, motherboard RAM size (for the 256K and 1 megabyte IIGs), terminator power (whether the card or drive will supply power to the SCSI bus termination), and ROM disk "visibility" (once the card is installed, you can "hide" the ROM disk so it doesn't continually show up in your drive list). Once the switches are configured, you can install the card in your computer, attach the drive, and move on to preparing the drive for use.

The configuration program provides for formatting and partitioning the hard disk; it also arranges to save your current settings to the hard disk (the program configuration is stored in the second half of the SCSI drive's physical block 0, which is normally unused and is invisible to the Apple II operating systems).

You probably will want to back up your current hard disk and reformat it with the *RamFast* utilities since the *RamFast* is capable of keeping up with a 1:1 interleave of blocks on the hard disk (that is, consecutive blocks on each track of the hard disk are consecutively numbered). With slower cards, the block numbers on a track are sometimes "interleaved" in other than strict ascending numerical order so that the interface has some extra time to process the previous block's data while waiting for the next consecutively-numbered block to spin into place. The *RamFast* is quick enough to read the blocks without requiring this delay; its design includes a 256K RAM cache to buffer data on the interface and a Z-80 compatible microprocessor to intelligently manage the *RamFast's* data transfer between the drive and the host computer.

The primary limitation of the *RamFast* is that it is more limited in its "true SCSI" device support. CD-ROM is currently not supported, and removable-media devices like the SyQuest may be a real problem (imagine ejecting the disk before the *RamFast* has a chance to write its data!). Arbitration (now supported on Apple's card), where the interface can defer to another host computer that may be accessing the same SCSI device, may also be difficult to support. On the other hand, *RamFast's* use of only a subset of the SCSI standard also allows it to use SASI devices such as the older First Class Peripheral's *Sider* drives. C. V. Tech has also expressed the intention to work on enhancing device support.

There has been a running argument on GEnie about which card is better. Apple's card does not include the RAM cache and on-board intelligence features of the *RamFast* and the associated speed advantage (the Apple card achieves some speed increases by using synergistic features in GS/OS). It does appear to support the widest range of devices currently, however. The *RamFast* appears specifically tailored to hard disks with non-removable media, and if performance is the primary consideration it wins hands-down with current software.

The new rash of DMA SCSI interfaces has also raised the issue of DMA compatibility again, and the presence of these interfaces has allowed us to test the RAM cards we use here. The compatible cards we have found are the ones that use only four rows of RAM chips: the original Apple (one megabyte) *IIGs Memory Expansion Card*, the Ingenuity *GS Juice*, and the Chinook *GS-4000* memory cards. The Cirtech *GS-8* does not work with the DMA option enabled; a fix is available from Cirtech but it will eliminate the use of more than 4 megabytes of memory on the card.

We have been told that the current 8-bit Apple II accelerators are not compatible with these interfaces in the DMA mode. The anticipated Applied Engineering *TransWarp III* is expected to be DMA compatible; the *TransWarp GS* already is (if you have an older card, you may need to verify this with Applied Engineering).

Other types of cards that may use or interfere with DMA are video digitizers, coprocessor cards (a specific example is the original Microsoft Z-80 *SoftCard*), and interface cards that allow interrupting the processor (such as the ThirdWare *Fingerprint*). We don't have a

comprehensive list, so the standard trial-and-error method of finding the offending card applies: with power to the system off remove all unnecessary interfaces, then add them back to the system one at a time (powering up each time to test) until the problem manifests itself. The last card you added before the failure is the likely culprit.

If you have a problem with using the interface with DMA enabled (it should be dramatically evident; our cards crashed while attempting to boot GS/OS), try flipping the DMA switch on the interface off and see if the card behaves. If so, another device in your system is probably not DMA compatible and you'll need to contact the manufacturer of the likely card to see if a cure is available.

We've noticed a recent rise in the instances of user group newsletters reprinting A2-Central copyrighted material (above and beyond the "fair use" exceptions) without previously asking for and being granted permission. We'd like to gently remind the respective editors that this is a "no-no"; we've even sent a few written reminders. The most common misunderstanding seems to be that if we grant permission to reprint once that the same procedure can be used for a future article without a separate request. Each instance and each publication requires specific, separate permission. We are generally agreeable to occasional reprint requests, but the protocol must be satisfied.

It's also healthy to have as much diversity in user group newsletters as possible. While imitation may be considered flattery, originality adds to the information pool.—DJD

Windows and Icons and Mice

Last month we looked at Apple's ProDOS 8 selector and the use of BASIC.System as a selector. This month, we'd like to examine Apple's standard IIGS selector tool: the Finder.

The IIGS Finder functions as a disk utility (copying, initializing, renaming, and so on), a file utility (copying, deleting, renaming, and so on), and as a program selector. Finder's basic operation is covered in your IIGS manual (for newer systems), or in the documentation which accompanies the IIGS System Software user packages.

Finder generally uses graphic symbols called *icons* to represent devices and files. In most cases, the entity (disk volume, file, or symbolized function such as the Trash) you wish to work with is chosen by moving the mouse pointer to the object's icon and selecting it with a click of the mouse button; Finder then "highlights" the icon to indicate that it is subject to manipulation by commands you select from the menu options or via the quick-key equivalents. Finder is easy to use when selecting among a few files on a floppy disk, but becomes less "friendly" as the number of files increases. On a hard disk, the Finder can start to flounder in a maze of windows.

Finder's program selector functionality is tied to the ability to define attributes of individual icons to specific file (or device) names and types. When you double-click (by pressing the mouse button twice in quick succession) on an icon, Finder will take the action it deems appropriate by referring to the icon's attributes. For disk volumes (usually represented by a disk drive or diskette icon) and directory files (represented by a "file folder" icon) Finder will attempt to open and display the contents of the volume or directory. For an icon representing a program, Finder will attempt to execute the associated file.

The fanciest trick is performed when the icon represents a data file associated with a specific program, such as an AppleWorks GS word processing file. The icon definition can include a reference to the application program that uses the file. If you double click on such an icon, Finder will attempt to launch the application program and pass the name of the selected file to the program via a function of the IIGS's Tool Locator tool set called the Message Center (described in the *Apple IIGS Toolbox Reference*). If the application is prepared to extract the name from the Message Center, it can immediately proceed to processing the selected file as directed by the message type. If multiple files of the same type have been selected for the application, Finder will pass all of the names in the hope that the application can deal with them (if the set you've selected can't be dealt with by Finder, you'll receive a "Those items can't all be opened/printed at once" error message). Currently, commands to "open" or "print" files are the only messages the Message Center has available to Finder.

Icon appearance and function. The second volume of the *Apple IIGS Toolbox Reference* defines an icon (in Chapter 17, regarding the QuickDraw II Auxiliary toolset) as "a small graphic object that is usually symbolic of an operation or a larger entity, such as a document". QuickDraw II Auxiliary also defines an icon in terms of its physical identity to a program: a specific data structure called an icon record that QuickDraw II uses to construct the image you see displayed (Figure 17-1, p. 17-3 in the *Toolbox Reference*).

The IIGS icon definition provides for the description of the icon image itself, and also an icon mask that can be used to "filter" the appearance of the image as displayed. There is also a *displayMode* word which determines how the mask is to be applied to the icon. A *displayMode* of 0 indicates the icon is to be copied directly through the mask, that is, for every point in the mask, the corresponding point in the icon will appear on the screen. At the points where there is no mask, the background will be displayed instead. The other *displayMode* settings provide for shading the icon in various ways; Finder uses these to "dim" icons belonging to a disk that is no longer on-line, as an example. (For an example, peek ahead at Figure 1.)

Finder's icon files (type \$CA) contain definitions that coordinate each icon image to a class of objects (devices or files). In addition to the icon record, each definition includes the file type (other devices also have a "type" assigned) and auxiliary type of the file(s) the definition is intended to apply to, and an application pathname.

The tool needed to customize Finder as a selector is a Finder icon file editor (we're just going to use the term "icon editor" from now on, but it should be realized that we're talking strictly in terms of an editor designed for Finder icon files). An icon editor can be used to alter an existing definition or create a new definition.

Icon editors have appeared as shareware: Dave Lyons's *DicEd* (\$15 shareware from DAL Systems, P.O. Box 875, Cupertino, Calif. 95015-0875; this program was included on our May 1989 monthly disk) and Paul Elseth's *Icon Editor* (\$15 shareware from Paul at 2739 Fairview Ct. S.E., Rochester, Minn. 55901). In addition, Apple has released their *Apple IIGS Icon Editor v1.1* (\$25) through APDA (#A0015LL/A) in a beta version, but we found *DicEd* and *IconEd* to be more usable.

DicEd and *IconEd* are very similar in function, with the exception that *IconEd* displays a window containing the icon images and allows you to select the icon you wish to modify from those in the window while *DicEd* displays the icons one at a time as you step forward or backward through a set. Both programs allow you to move icons within a file or between files by "cut and paste" methods, but *IconEd* always pastes the new icon at the end of a list while *DicEd* allows you to paste the icon into the file at the position following the icon you're currently viewing. This makes selecting an icon to edit easier with *IconEd*, but changing the order of the icons easier with *DicEd*.

These editors let you draw or modify the image for the icon itself, both in a "large icon" and "small icon" format. Drawing or modifying the icon image is performed much as you would draw any object with a paint program. The editor allows you to select the size you intend to use for the icon (large icons may overlap "standard sized" icons when placed using Finder's default "grid", so moderation is advisable), and then to plot the image you want to use. You also have to construct the mask you wish to use with the image; in simple cases it may be adequate to copy the image to the mask, but if you want to see the background of the desktop rather than the background the image was drawn on you'll need to modify the mask accordingly. Figure 1 is a sample of *DicEd*'s display as you edit an icon; notice the sample *displayMode* alternatives shown above the icon image and mask which are being edited.

In addition, *DicEd* and *IconEd* allow you to save the icon definition in a source code format that can be incorporated into a program; this allows the programs to function as more general-purpose editors to create icons for other applications.

Selecting the type of file. *DicEd* and *IconEd* allow you to select the file type (and auxiliary type) from a list of "common" types. If the filetype you wish to use is not among those listed in the editor, you can enter the type manually (as a hexadecimal number) into a "file-type" field within the editor. For example, if you wish to define a specific icon for GS/OS source files (type \$B0), you enter "\$B0" as the

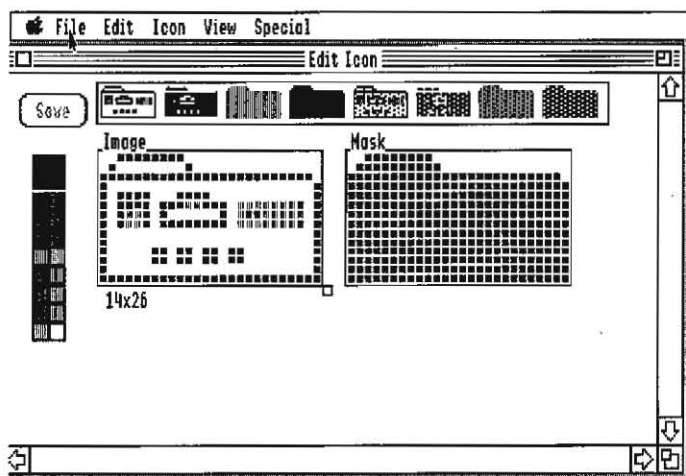


Figure 1. Icon, mask, and displayModes

file type. The file type for a GS/OS application is \$B3.

There is also a provision to enter a specific auxiliary type value, and you must be more careful with this field. For most programs, you should leave the value at "\$0000", which will match all auxiliary types for the specified filetype. This is especially important for files that use varying auxiliary type values, as AppleWorks (which uses the auxiliary type field as a "case mask" for the filenames it displays); if you enter a specific auxiliary type, it will be the only one that Finder will associate that icon with.

Finally, there is the option of selecting an application pathname. This is the key to Finder's use as a selector; assigning the proper application name to a file's icon allows Finder to locate and launch the proper application automatically when you "open" the file.

You should set the application pathname for the icon to the full pathname for the location of the program file that would expect the selected file as data. For an application itself (such as *AppleWorks* GS), there is no need to enter such a pathname. For *AppleWorks* GS specific files, you can enter the pathname of the *AppleWorks*.GS file on your disk.

A note here: formerly, under ProDOS 8 and ProDOS 16 (before GS/OS), many file types were given a three-letter abbreviation, such as "S16" for 16-bit applications. It is becoming necessary to expand the number of available types available beyond the 256 combinations available to the one-byte filetype, so in addition to the filetype Apple is now arbitrating and assigning auxiliary types for files. Officially recognized uses are delineated in the *Apple II File Type Notes* distributed as part of the *Apple II Technical Notes*. For example, a GS/OS source file for the *APW C* compiler is assigned an auxiliary type of \$0005, so that could be entered into an icon definition specifically for *APW C* source files. The use of three-letter abbreviations is becoming awkward, and the list won't be expanded to try and encompass all new filetype and auxatype designations. Instead, Apple is assigning more descriptive names such as "GS/OS application" to be displayed by Finder. To get the filetype and auxiliary type of a file, you may need to use a file utility like ProSel that lists the raw types. Current IIgs programs may include icons to be installed with the program; if so, you can save the effort of trying to determine the types and just modify (your installed copies of) the existing icons.

Gathering together the icon files. In order for Finder to function, it must be able to locate an Icons folder in the root directory of the disk you boot GS/OS from. The standard Finder icons are located in two files: *Finder.Icons* and *Finder.Icons.X*. The primary (and required) *Finder.Icons* file contains the minimal system icons that Finder might reasonably need to use; the 3.5 and 5.25 drive icons, hard disk icon, trash can icon, and several common file icons. *Finder.Icons.X* contains "expendable" icons that represent objects that may be less common; icons for AppleShare, CD-ROM, and the like (occasionally, this file may be removed by Installer if disk space is tight).

In addition to the Finder icon files, Finder now also uses filename descriptor files to hold those descriptive names for the icon's object, such as "GS/OS application" for a file of type \$B3. These files are

named *FTYPE.MAIN* (required) and *FTYPE.AUX*. The GS/OS Installer will place these icon files on a disk when you install or update the GS/OS system files.

You can add other icon files to the boot volume's Icons folder, or place them in an Icons directory in the root directory of another disk volume. When that disk is placed on-line, its icons will be used to augment those icons Finder has already culled from other disks already on-line. Finder retains the icon information in memory so the icon can continue to be displayed even if you eject the disk. When you exit Finder (to run an application, or via "Quit") the images in memory will be lost and Finder will build them again (from any available volumes) when it restarts.

The system is not insensitive to problems; some of us have seen "disappearing Trash cans" and other artifacts of icons gone wrong. Apple's Rob Moore and Finder guru Jim Merritt have circulated a document with several suggestions for avoiding clashes, among them:

*Don't modify the standard Apple icon files *Finder.Icons* and *Finder.Icons.X*.* Instead, load the file with the icons you wish to modify (which could be a copy of the *Finder.Icons* or *Finder.Icons.X* file), delete all icons you don't intend to use (and specifically make sure you delete any "generic document" icon so it won't conflict with the one in the *Finder.Icons* file), then modify the icons of choice. Then save this new file under a new name (that is, add the new file, leaving the original *Finder* icon files intact). You can modify the Trash icon's image but be cautious of increasing the size and don't change its other attributes (such as "type").

Don't omit the filetype when using "wildcard" names to assign the icons.

They also confirm that (other than *Finder.Icons*, which is loaded first), *icon files are loaded in directory order within a directory*. If multiple icons could match a file's characteristics then the last icon loaded that matches will be used (this is the reason for avoiding deleting or re-defining the "generic document" icon in the *Finder.Icons* file). If you see an icon that does not match the one you thought you had defined, search through your icon files for a (later) duplicate icon. (The search order will also be affected by the order in which the disk volumes are "scanned" as Finder looks for new on-line volumes.) From our experience, icon files from a volume will be ignored if they duplicate the name of a file that has already been read.

As an example, the Drivers folder used in the icon editor example was defined with a filetype of "\$0F" (folder) and a name of "Drivers" (auxtype remains \$0000, and the application path was left blank). When a folder of the name Drivers appears as an icon, the edited image is displayed rather than the default *Finder.Icons* "folder" icon since the new icon definition takes precedence. Additional icons use memory and disk space, so you may want to be conservative in any augmentations.

With your *Finder* icons file prepared, you're ready to set up your selection "menu"; the *Finder* desktop.

Installing programs on a large volume. In most cases, unprotected, correctly programmed ProDOS and GS/OS applications can be placed on a hard disk. Our normal procedure is to create a new directory for the program and then copy all of its files into that directory with the exception of any common files that may already exist on the hard disk (such as "PRODOS" and "BASIC.System"). IIgs (or ProDOS 8) applications that require more than this simple installation should either have an installation utility provided or more detailed instructions in the manual. If you have questions or problems, let the manufacturer hear about it.

A hard disk of any size can quickly gather hundreds of files. Remember that the ProDOS disk format that GS/OS also uses can only support a maximum of 51 names in the root directory. To add more files than that, you will need to start creating and using subdirectories to hold the additional files. Before loading up a hard disk, you should sit down and try to plan how you want to organize it.

Since the number of files allowed in the root directory is limited, we tend to make many of those files subdirectories. For example, I have subdirectories on my hard disk named "Aps" (miscellaneous applications), "Comm" (communications), "Dev" (development), "Graphics", "Sound", "Utilities", and so on. Each of these

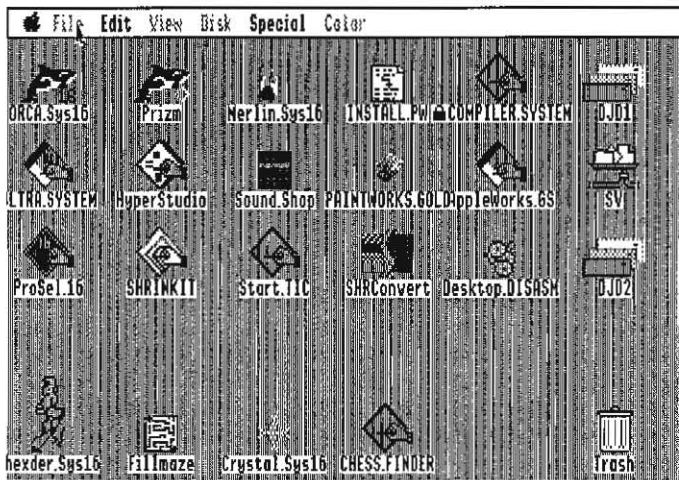


Figure 2. Gridlocked desktop

"categorical" directories contains several directories with individual programs; "Dev" contains folders for "Merlin16P" (*Merlin 16 Plus*), "Orca" (*ORCA/M* and tools), "APW" (*Apple Programmer's Workshop* and tools), "AS" (Applesoft), and so on. Most of my actual executable programs are therefore about two directory levels deep. If you have fewer programs, you may wish to keep them only one subdirectory level deep (that is, you might prefer to have an "AS" folder in the root directory and skip the "Dev" folder).

Once you have the organization firmed up, you can work your way through your Finder icons with an editor and start defining their application pathnames. We mentioned last month that ProDOS 8 allows you to define a prefix path that it "remembers" as the default pathname to use for access to a specific directory. GS/OS supports the use of 32 prefixes (numbered 0-31) as well as a defined prefix of "" to signify the name of the boot volume. If your programs reside on the boot volume of your hard disk, you should use "" as the volume name when defining the application pathname; in that case if you should happen to rename your boot volume Finder will still be able to locate the proper application. For example, on my hard disk named "/CT30" I have the path to *ORCA* defined as ""/Dev/Orca/Orca.Sys16" ("" is interpreted by GS/OS as the name of the boot volume, so the prefix expands to "/CT30/Dev/Orca/", and "Orca.Sys16" is the name of the system file).

The intermediate part of the application path will change if you rename or shuffle the location of your subdirectories, which is why it's important to decide on some "standard" way of organizing your hard disk. Every time you rearrange things, you'll need to work your way through your icon files to check the application pathnames. Additionally, if you boot from a device other than the hard disk on which your applications reside, you probably will want to hard-code the name of the device into the application pathname so that the hard disk (rather than the boot device) will be accessed. That is, you'll want to use "/CT30" instead of "" in defining the application path so that Finder will always look for the volume /CT30. Note that if you do this, you'll have to edit the application pathnames if you rename the hard disk. Also note that if the application disk isn't on-line when you click on one of its data file icons, Finder won't prompt you for the missing application disk; it simply reports the application can't be found.

Using the desktop to advantage. Most of the awkwardness of using Finder comes from having to open and close multiple windows to access files that are not readily accessible. With a large storage device containing many files, Finders "ease-of-use" drops off drastically as you have to open and keep track of your location within several windows. There are ways to combat this syndrome.

The first strategy is to have the correct icon definitions for all the files you commonly use. If you can click on an icon as soon as you locate the file of interest and have the proper application automatically launched (rather than having to scour your hard disk for it), a lot of work is eliminated.

The second strategy is to make commonly used files as accessible as possible. The IIGS Finder allows you to drag an icon (other than a

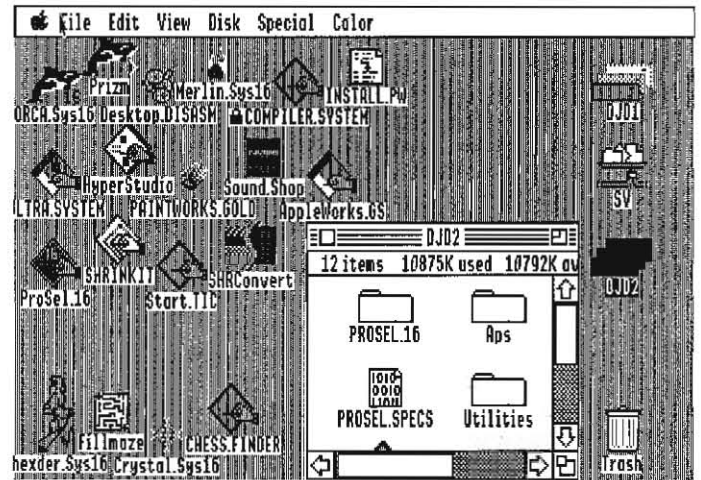


Figure 3. Compacted desktop

folder) out of its window and onto the desktop itself; doing this allows you to launch the application by double-clicking on an icon without having to open a window. Unlike the Mac Finder, the IIGS Finder is smart enough to remember what folder the icon actually resides in (and if you select "Put Away" from the IIGS Finder's "File" menu with an icon selected, it will be put back in its correct folder). Dragging icons for commonly needed files onto the desktop allows you to access the file without requiring you to have any windows open, making the desktop less cluttered. One annoyance that you have to watch out for is that if you copy or move a folder to another folder or disk with Finder, Finder will "Put Away" any icons from that folder that have been dragged to the desktop. You'll have to find them and drag them out again.

Is your desktop getting crowded? *The Macintosh Bible* (a great reference; we long for a IIGS version) has a clever suggestion; if you "stagger" icons so that the names and icon images interleave slightly, you can get many more icons into the same space. Using the "Clean up" option of Finder's "Special" menu to arrange the icons, we ended up with a desktop as shown in Figure 2. Staggering the icons (which is easier to show than it is to explain) gave us enough room to add an open window to the Desktop as in Figure 3.

If you want to automatically keep the icons and windows where you put them, be sure to set the preferences (under Finder's "Special" menu) so that the Finder information is saved to disk in those Finder.Data files.

There are alternatives to Finder. When GS/OS boots, it looks for a program named Start in the System folder of the startup disk. (If Start isn't found, GS/OS starts scanning the root directory of the startup disk for the first S16 file with a name ending in ".SYS16" or the first SYS file with a name ending in ".SYSTEM".) As the IIGS System Software is supplied by Apple, Start is actually the IIGS Finder. There are alternative third-party selectors, though.

Squirt-16 (\$20 shareware from Synesis Systems, P.O. Box 1308, Gilbert, Ariz. 85234) is a text-based selector that will launch 16-bit and 8-bit system programs as well as Applesoft BASIC, binary (BRUN), and EXEC program files. It will maintain a "run list" of 10 programs for quick launch, and will let you set up a program to automatically run on startup. It includes a help screen, and requires ROM version 01 or higher.

Command line driven shells are available; as for ProDOS 8, many are supplied as part of a programming environment, such as the shells supplied with the *Apple Programmer's Workshop* from APDA (Apple Computer, Inc., 20525 Mariani Avenue, MS:33G, Cupertino, Calif. 95014, 408-562-3910) and the Orca languages (assembler, C, and Pascal) from The Byte Works, Inc., 4700 Irving Blvd. N.W., Suite 207, Albuquerque, N.M. 87114, 505-898-8183. The shells can be programmed by writing text files containing shell commands, and new external shell commands can be written and added to the environment. A special type of file (GS/OS Shell application, sometimes

abbreviated as EXE) exists which is intended to be run specifically from a shell environment.

ProSel-16 (\$60 from us, and from the author Glen Bredon, 521 State Road, Princeton, N.J. 08540) combines a menu-driven selector similar to *ProSel-8* with a programmable shell that can be accessed directly from any menu. Recent versions allow the selection of text-based or graphics-based menu screens, and give access to the desktop for use of New Desk Accessories (from the "apple" menu). *ProSel-16* allows launching S16, EXE, SYS, and BAS (with options such as startup pathname specification, purging options to attempt to free "idle" memory for large GS/OS applications, and IIGs slow/fast speed settings) files as well as its shell command files.

There are alternative desktop-based selectors. One of the simplest is the freeware *LaunchBox*, which allows you to define up to 23 buttons (via a text file) that can launch applications. A twenty-fourth button is reserved for launching other applications for which a specific button hasn't been defined. Making multiple menus involves creating multiple copies of *LaunchBox* and its application lists. It cannot pass a startup pathname, or set an application path distinct from the path the application is launched from.

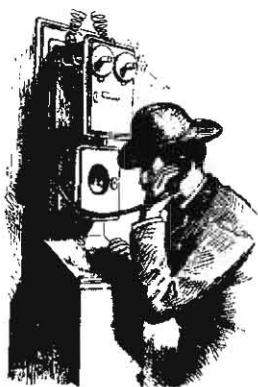
HyperLaunch (\$60 from So What Software, 10211 Slater Avenue, Suite 103, Fountain Valley, Calif. 92708, 714-964-4298) also uses the button metaphor, and also includes file and disk utilities.

JumpStart (from Dave Hill at MaineFrame Software, 255 Mitchell Road, Cape Elizabeth, Maine 04107) is a program selector with a

"quick run" list and file and disk utilities. *JumpStart* can be driven by a text script, and includes provision for an application to be executed at startup. If you don't need (or have room for) the disk utilities, a more compact version of the *JumpStart* program without the disk utilities is provided.

When using an alternative selector, either it can replace Finder as the Start program by deleting Finder, or it can rename the current Start file to something else, or it can be installed to be launched from Finder. Unless disk space is at a premium, we've generally chosen to keep Finder available by renaming "Start" as "Finder" before installing a new selector. Before you remove Finder, make sure that your alternative utility has enough options to allow you to perform all the normal maintenance functions you may need; access to New Desk Accessories (for example, the "Choose printer" option now is only available in the Control Panel NDA!), ability to activate/inactivate initialization files (Finder allows this from the Icon Info dialog for an icon), controlled Shutdown (available from Finder's "Special" menu) and so on.

For AppleShare, we find ourselves using Finder because its utilities are fully AppleShare aware, and Finder's Icon Info dialog for file icons residing on an AppleShare volume allows changing access privileges (the IIGs AppleShare interface is superior to the Mac interface in this respect).—DJJ



Ask (or tell) Uncle DOS

24-pin printer support

I have some information in response to Rick Pedley's question on the Panasonic KXP1124 24-pin dot matrix printer, which appeared in *A2-Central* Vol. 6 No. 3, page 6.25.

The Panasonic KXP1124 is electronically and mechanically identical to the Roland *Raven* PR2417, except for some of the external casing. This was confirmed by careful examination of the technician's service manual for the '2417, which uses the two printers' names interchangeably.

I purchased the Roland *Raven* PR2417 last June. According to the self test, its ROMs are revision level "C". I use it on my Apple IIGs (ROM version 01) with an Orange Micro *Grappler Plus* (ROM revision 3.2) parallel printer interface in slot 1.

The '2417 has basically two emulations: Epson LQ2500 and IBM *ProPrinter* X24. Most printer control commands are common between these two printers (i.e. they have the same effect), but there are some very important exceptions.

Problems arise because some bit-graphics modes are not shared (e.g. Epson's "Esc ? n m" command), and that the same linefeed commands ("Esc A n" and "Esc 3 n") do different things. In Epson mode, the former sets paper feed to n/60 of an inch and the latter sets the paper feed to n/180 of an inch. In IBM mode,

the same commands set paper feed to n/72 and n/216 respectively.

Some software explicitly supports Epson LQ-series printers; for example Print Shop GS, Publish-It! 3, Graph-It!, TimeOut SuperFonts, etc. These programs will work properly if they are installed for the Epson LQ. However, most programs (particularly older ones) only have support for 9-pin Epsoms (e.g. the MX, LX, FX, etc.). If software is installed for an Epson FX80 and the printer is in Epson LQ mode, the '2417 will do a good job of emulating that 9-pin Epson, except that all bit graphics printouts will be stretched vertically by approximately 20%. This is because the Epson FX's linefeed value is not the same as the Epson LQ's. The solution, of course, is to set the '2417 to IBM mode, and then the aspect ratio will be preserved and there is complete compatibility. About 99% of software I have encountered falls into this category.

However, a problem arises when a program (installed for an FX80) sends bit-graphics commands which are not supported in the IBM mode. The result is seemingly endless pages of "garbage". The only example of this is the Activision GS/OS Epson driver. In this case, there is no choice but to set the '2417 to LQ emulation, and suffer with slightly stretched printouts.

For text-based programs (e.g. AppleWorks) there are no problems as long as the '2417 is in the mode the program is installed for.

Other than this one minor nuisance with graphics, I am completely happy with the performance of this printer. The printouts are razor sharp and uniform, and the paper transport system is very versatile and reliable.

Stephen Brown
Willowdale, Ont.

No zip in memory card

Ever since I installed a *Zip Chip* in my Apple IIc, I keep having unexpected crashes in AppleWorks. What's the deal?

James A. Strauss
San Francisco, Calif.

Chris Adams at Chinook says they have seen

problems with the combination of the *Zip Chip* and Apple's internal memory expansion for the Apple IIc. This seems reasonable, since Apple recommended against using their expansion in the IIc Plus due to compatibility problems with the accelerated CPU.

Chris says Chinook can modify the RAM expansion card for compatibility for the bargain price of \$25; contact Chinook Technology, Inc., 601 Main Street #635, Longmont, Colo. 80501, 303-678-5544.—DJJ

Bad driving?

Rumor has it that third party disk drives that would work with the old IIGs will not work with the new (ROM 03) version. Comment, please.

Bill Gallip
Virginia Beach, Vir.

The new second revision of *Apple IIGs Hardware Reference* does show a slight modification (mostly added resistances) to the disk port hardware. We have seen one anomaly connected with this; under GS/OS, if the operating system is scanning for a disk and reaches the AMR 3.5 drive we have installed on one of our ROM 03 machines, GS/OS will "hang" if the drive is empty. Inserting a disk in the 3.5 drive allows GS/OS to continue the scan.

Another anomaly we have heard about but haven't seen ourselves. A technician contacted us to tell us that there seems to be a problem with many third-party 5.25 drives when attached to the ROM 03 disk port daisy chain. His point was that Apple drives use the Phase 1 stepper motor (the device that positions the drive's read/write head over the proper disk track) signal as the write protect signal return while most third party drives use a different connection. His explanation of the problem is that a non-write-protected disk placed in the 5.25 drive can't be accessed by the ROM 03 IIGs (write-protecting the disk will allow read access).

The question is whether Apple made the change in the disk port in order to improve

performance for their products, or whether Apple made the change to intentionally eliminate some third party products. While third-party compatibility shouldn't need to be Apple's first concern if they are moving to improve the reliability of their own products, intentionally making a change to obsolete a user's equipment is not very nice (and customers tend to remember such things).—DJJ

AppleWorks file names

When booting GS/OS 5.0 to Finder then going to AppleWorks classic I would receive error messages trying to load certain files to the AppleWorks desktop. Verifying those files using the Finder showed no errors. This problem seems to be OK booting 5.0.2...what gives?

Kirk Klemme
Marquette, Mich.

We're not sure why you would get this error with System 5.0, which includes ProDOS 8 v1.8, unless you are booting AppleWorks from a disk using an earlier version of ProDOS.

ProDOS 8 v. 1.7 wasn't prepared to handle the altered *min_version* numbers that GS/OS now uses to simulate lower case in filenames. A ProDOS 8 program could then receive an error while trying to load a file using a pathname with (System 5.0) "lower case" in the pathname. We ran into this trying to load files with AppleWorks from a directory which had been named in mixed case from GS/OS. Watch the screen carefully as you launch AppleWorks and see if you can catch the ProDOS 8 version number shown on ProDOS 8's copyright screen.

AppleWorks uses a different mechanism to designate the lower case characters (and spaces) it uses in filenames, so the "lower case" characters defined by GS/OS's mechanism won't be visible as lower case within AppleWorks. This added to our confusion. Moral: always use the most recent version of the system software.—DJJ

Image processing

If the "G" in GS stands for "graphics", is there a way to do image processing on the IIGs? The book *Beyond Photography, The Digital Dark-room* by Holzmann (Prentice-Hall, 1988) lists a program in C language that seems to do just what I have been wanting my IIGs to do. Is it possible to use this program on the IIGs?

Bernard Dubois
Montreal, Que.

I'm not familiar with the book you mention, but digital image processing is essentially a mathematical operation on a set of data, and that ability falls within the range of any computer. The limitations on this processing are data capacity and speed; a large data set or the need to carry out a lot of manipulations may force you to a larger system. Digital image techniques used in commercial films such as Disney Studio's *Tron* involve the use of the most powerful computers currently available. But "Part 2 of A Beginner's Guide to Spectral Analysis" (*Byte* magazine, March 1989, pp. 166-198; Part 1 in February 1981 pp 68-90 gives a sample of analyzing of a "musical spectrum") contains an example of image processing using 6502 assembly language and a Commodore microcomputer that could be adapted to any Apple II system (you will need to know how to access the Applesoft ROM's floating point math routines to convert the program).

The other limitation is the display; if you want the system to render a completely "life-like" image, you will likely be disappointed since most personal computers within the normal budget do not have such graphics capabilities. But don't despair; even professional development is often prototyped on systems of limited resolution.

The portions of the programs you have found that manipulate the graphics data should be adaptable to the IIGs. Like many such programming questions we receive it's not practical for us to try to tell you exactly **how** since that involves dissecting and analyzing how the program works, and that is in your hands. The portions of the program that manage the display of image are probably hardware dependent (that is, they require knowing how the particular computer in use is physically building the "raw data" into the display image), and that portion will need to be modified to take the specific operation of the IIGs display graphics into consideration. Fortunately, Apple's technical manuals give the details; the lowest level considerations of the graphics display are covered in the Apple IIGs Hardware Reference, and access to higher-level drawing tools are covered in the Apple IIGs Toolbox Reference. Be prepared to do some reading; the IIGs graphics tools QuickDraw II and QuickDraw II Auxiliary form a significant portion of the IIGs toolset.

If you want something more ready-made in nature, we received a program called **Second Chance v2.0** from John R. Majka at Raptor, Inc., PO Box 20756, Louisville, Ken. 40250, 502-491-6828. This is a IIGs desktop-based collection of nineteen routines for image enhancement that operates on 320 mode grayscale images. The routines take anywhere from a few minutes to a few hours to run.—DJJ

Too few characters

I would like to know if there is a program or patch available for AppleWorks that will allow me to enter more than 76 characters of text in an AppleWorks database field.

Paul Willis
APO New York, N.Y.

We don't know of a patch to extend a single category's field length. You may need to switch to a different database; one product we've mentioned in the past is **DB Master**, which allows up to 256 characters per record.

If you want really large fields, one option we know of is **ProFiler**, which has more of a "free form" capability. After our query last month, several readers contacted us to let us know that sales and support for the **ProFiler 4.0** database program has reverted back to Custom Computer Products, 355 Costa Mesa Street, Costa Mesa, Calif. 92627, 714-548-5429. The retail price for 4.0 is \$129.95.

ProFiler is a MouseText-based database that supports up to 65000 records containing up to 250 fields. It allows you to define forms made up of fields which may be of text, dollar, formula, link, launch, or graphic types.

The text field will support word wrap (via an control-W command) if your data comprises more than one line.

Formula fields use a mathematical formula to calculate a value based on the contents of other fields.

Link fields allow you to link a second database file to your open **ProFiler** file. You

create the link by entering search criteria as "data" into the link field when the field is defined (the search criteria applies to the first indexed field in the secondary file). When you select the link field with open-apple-L while using the data file, **ProFiler** will use the search criteria to bring up a summary listing of applicable records from the second file. You can scroll through this listing and select the record from the second file you want to view. Selecting "Close" or "Quit" from **ProFiler**'s command menu returns you to the primary file.

Launch fields allows you to enter the name of a (ProDOS 8) application. When the field is selected and activated by open-apple-L, **ProFiler** will launch the application. When you quit the application (if the standard ProDOS "quit" mechanism is used), you will return to **ProFiler**.

Graphic fields can be used to display Apple II high-resolution or double high-resolution graphics on 128K systems, and super high-resolution graphics on an Apple IIGs.

ProFiler also includes macros (up to five macros of 32 characters each) that can be stored with the database, standard text field data ("fixed fields"), and most standard database operations (editing, searching, printing, and so on) and includes a utility for creating mailing labels (up to four across a page). **ProFiler** requires a 128K enhanced IIe, IIc, or IIGs; it supports a mouse, and a large capacity disk drive (3.5 or hard disk) is recommended.—DJJ

Business Card driver

In response to a user in a recent issue of **A2-Central**, I've just uploaded a program to GEnie that will install a clock driver for the Street Electronics *Business Card* clock into ProDOS 8. This driver is version independent, and should work with all past and future versions of ProDOS 8 without requiring the modification of ProDOS itself. The driver is installed onto the users disk by an installer program, and becomes the first ".SYSTEM" file on that disk. It's called BC.CLOCK.SYSTEM. If it finds the *Business Card* clock, it will install the driver inside ProDOS memory where the clock driver is supposed to reside. If it doesn't find the *Business Card*, it doesn't install the clock driver, and just continues on with the boot in a normal fashion. It does look for the *Business Card* to be in slot 1 of the user's computer. Cheers.

Craig Peterson
Santa Monica, Calif.

Thanks, Craig. We don't expect this to be a high-volume item, so for those who don't have access to GEnie but have a need for the driver, send us a blank ProDOS-formatted disk with return postage and we'll get a copy to you.—DJJ

Applesoft floating point

Do you know of a book or other source that explains, preferably with examples, binary floating point arithmetic as it is used in the Apple IIe? In my study of ProDOS, I wrote a simple program which assigns values to both integer and real variables and strings to string variables. The program then listed out the contents of the memory locations containing the program itself and the variable table. After consulting the

books I had, I was able to translate the addresses contained in the string variables and the values in the integer variables. I even managed to translate the values of the real variables when they contained only whole numbers; however, when they had been assigned values in the program that included decimal fractions, I couldn't translate the values shown in the variable table.

At the moment, I'm more interested in being able to decipher the decimal values contained in things like the variable tables of programs than I am in being able to perform arithmetic operations on floating-point numbers. However, I'll probably get around to the latter, eventually.

Incidentally, I have a copy of the *Apple Numerics Manual*. It's apparently intended for mathematicians, not hackers. If it explained how to translate the data given by the variable table for real fractional variables I couldn't find the explanation.

Robert J. Hoyer
Kailua, Haw.

*The reference you're probably looking for is Chapter 8 of **Assembly Language for the Applesoft Programmer**. It explains the Applesoft floating point formats and also gives some details on accessing the Applesoft ROM routines from assembly language.*

*The **Apple Numerics Manual** describes the SAME (Standard Apple Numerics Environment) tools that Apple has made available by license for the Apple II and in the IIGS System Tools. The floating point formats are not the same as those used by Applesoft.—DJD*

A2-Central™

© Copyright 1990 by
A2-Central

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.

Edited by:

Dennis Doms

with help from:

Tom Weishaar	Sally Dwyer	Dean Esmay
Joyce Hammond	Jeff Neuer	Jay Jennings
Tom Vanderpool	Jean Weishaar	

A2-Central—fired **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): \$28 for 1 year; \$54 for 2 years; \$78 for 3 years. All back issues are currently available for \$2 each; bound, indexed editions of our first four 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 \$84 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 15 cents per page per copy distributed.

WARRANTY AND LIMITATION OF LIABILITY. I warrant that most of the information in **A2-Central** is useful and correct, although errors 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 unfulfilled 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 case shall I or my 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
Voice: 913-469-6502
Fax: 913-469-6507

Printed in the U.S.A.

Hard drive down under

I have just received my 80 meg hard drive and would like to share an interesting fact which would benefit readers from 220-240 volt countries (i.e. Australia and European countries). The power supply which comes with the Leadman case you sell requires no modification to run with their voltage. Originally I was going to replace the power supply with an old one from an IBM PC. Out of curiosity, I studied the existing power supply and found out that it can be switched merely by pulling out the black active wire from its 115V pin and plugging it into the corresponding 230V pin. All one now needs is a power cable with the appropriate pin formation (i.e. identical to the power cable attached to your Apple computer and monitor); this can be found at most electrical and computer stores.

An 80 meg SCSI hard drive in Australia costs about \$2000 and a 40 meg SCSI hard drive costs \$1350—I spent \$1250 on your hard drive. This includes sales tax imposed by customs and the cost of buying a new power cable. It really does take about five minutes to assemble—with only a screwdriver. I couldn't be happier with the product and would recommend anyone to buy one if they are in the market for a hard drive. The \$750 I saved is now going towards the cost of a *TransWarp GS* and some software to fill up the drive.

Anton Sare
Moonee Ponds, Vic.

We have been told that other power supplies, including those used in commercial hard drives, may be adaptable to 220-240V, 50 cycle power. You should definitely contact your drive supplier before making any modifications to see if it is safe or supported.—DJD

6502 schools?

Are there any colleges, tech schools, etc. which offer courses in 6502 programming? I myself teach BASIC and AppleWorks along with macro programming. I believe that the human factor and the appropriate addition of expertise at just the right place and time helps many over the hump. Where can I find such a living guide?

Greg Presnell
Taylors, S.C.

The microcomputer programming community is still a rather scattered group; it seems that most interactive programming assistance we've seen comes primarily over the modem via programming forums on the commercial on-line services. Local user groups and their BBS's sometimes also create gathering places for budding programmers.

Some local "continuing education" programs may try to conduct such a class if there is sufficient interest, and if they can find a willing and capable instructor. We have seen this occur for Apple BASIC and AppleWorks classes in our area, but we have to admit that machine language programming classes have so far been absent.—DJD

Response from Broderbund

This letter is in response to a customer's correspondence you printed in your April 1990 Volume 6, Number 3 issue. The title of the letter you printed is "Too protected".

We are currently re-evaluating our policies regarding copy protection. As a matter of fact, we've come a long way this last year in produc-

ing products that do not include floppy protection schemes.

Broderbund Software produces a wide variety of software packages for different uses (i.e. entertainment, educational, personal productivity, and business). On some products copy protection has been necessary in the past to ensure revenues for the creator of the program, as well as the company.

However, times are changing and so are our customers. We have already responded to the needs of our customers and are attempting to meet those needs by not protecting our Business line of products. All of the new Entertainment/Educational products will be using off disk copy protection. The new copy protection scheme will ask the customer to provide some information that can only be obtained from the program's reference material or documentation.

Currently our Productivity line, such as the *New Print Shop*, is now unprotected, both IBM and Apple versions. We have also recently updated *Ancient Art of War and War at Sea*, as well as the *Carmen* series, to be hard disk compatible (IBM versions only). Most of our Entertainment programs, however, will remain protected, but we have implemented off disk copy protection schemes.

Due to the extent of software piracy we feel that copy protecting our software is still needed. We are attempting to develop a form of copy protection that will be less of a bother for the customer.

Joseph W. Shoemaker
Technical Support Manager
Broderbund Software, Inc.
17 Paul Drive
San Rafael, Calif. 94903-2101

As we perceive it, the author of the original letter was concerned about the types of copy protection that made a program unusable as a user upgraded his equipment, and about the possibility of the lack of an upgrade path.

We ourselves have softened a bit on the issue of protection because it seems that some manufacturers will not produce products for the Apple II without protection, and in some cases having a product with the awkwardness that copy protection adds is more desirable than not having the product. Of course, if the product ever stops working because of the protection, the confidence that other products from that company are a good investment may be compromised.

*We think the above concept of using "off disk" or "manual-based" protection rather than flaky disk schemes is a way to inspire more confidence that programs will be compatible with future hardware and peripherals that users will want to adopt. We know that many compatibility problems blamed on Apple's improvements in their hardware and software have to instead fall upon manufacturers; we have a protected copy of the IIGS game *Thexder* that seems to run under 5.0.2 from a hard disk with no problems (it uses a "key disk" protection scheme; you insert the master disk when requested to complete launching the program). Producing compatible yet "protected" software is possible. Broderbund seems to be aware of user concerns and we hope they continue to evaluate alternatives to find solutions acceptable to their company and their customers, and to upgrade their Apple II offerings accordingly.—DJD*