

A journal and exchange of Apple II discoveries

Year of the rumor

At the 1989 World Science Fiction Convention, respected science fiction and fantasy author Orson Scott Card conducted a panel session titled "A Thousand Ideas An Hour". The purpose of the session was to take suggestions from the audience and use them to build a framework for a story. At every step the story, though fictional, had to exhibit a certain conformity to logical examination so it would be believable enough to sustain the reader's (and the author's) interest. Orson Scott Card moderated by serving as the "reality filter", asking the audience to comment about the plausibility of each premise, discarding those suggestions which made little or no obvious sense and encouraging the audience to pursue those that sounded interesting. I watched in awe, because the session was obviously making the members of the audience think better, in more detail. This not only makes better writers, but better, more perceptive, people.

This type of thought process should be used to evaluate rumors. Even though a rumor, by definition, is not provable, it should be subjected to a reasonable "reality filter" that takes known facts into account, and attempts to resolve any associated rumors in the most logical manner that fits the known data. Though 1989 has certainly been the Year of the Rumor regarding the future of the Apple II (and specifically the Apple IIs), most rumors surrounding problems with the Apple II series computers quite simply don't make it through reasonable reality filters.

In 1989 we've heard various stories ranging from a new all-powerful IIs that will undermine the Macintosh, to a powerful, inexpensive Macintosh that Mac proponents within Apple will use to persuade the company to drop the Apple II. Impetus for the rumors comes from the rather confusing state of Apple's actions over the past few years, and possibly (only possibly) from occasional "leaks" within the company. Mostly, they are helped along by the determination of each camp's (Mac and Apple II) "faithful" to seize upon any excuse to lambaste the other. Apple itself hasn't done a good job of trying to allay the antagonism, which has even divided some user groups.

Some of the most damaging rumor mongering has been among the media, where the "reporting" of rumors seems intent on creating the very effect that they speak to. An example is the December 1989 *MacUser* "report" that the Apple II division will be spun off with Steve Wozniak at the helm, and a Mac SE with 20 meg hard disk will be released for \$995. These "journalists" are not reporting news, they are repeating statements not clearly based on facts and then adding an editorial opinion as if the rumor was fact. If allowed to spread without contradiction, a "domino effect" results. In an October 9 column syndicated by the *Washington Post*, staff writer T. R. Reid admonishes those participating in a promotion by Apple (which allows schools to collect and use grocery tapes from participating stores as credit for purchasing equipment) to designate that their receipts only go toward the purchase of Macintosh computers in lieu of the "anachronistic Apple IIs".

Rumors have essentially one positive effect: if they actually are based in fact, they allow some foresight and planning. Without skeptical analysis, though, this "planning" may be based on someone's idle whim, and the effects can be devastating. In relation to commercial products (computers, software, and so on) the rumor, or even the announcement, of a new improved system may kill sales of a current system at a crucial time. Adam Osborne's and John Dvorak's *Hypergrowth* details how Osborne Computer was driven into bankruptcy by the pre-announcement of a vastly improved computer (the Osborne Executive); the company was caught between lagging sales of the older Osborne 1 while not yet able to produce enough of

the new systems to meet orders. Another example is the market-wide slump which occurred before the introduction of the IBM PCjr as buyers waited to see what IBM would offer (it seems laughable now, of course). For this reason, Apple and many companies have adopted the "vow of silence" that IBM has long held: "we do not comment on unannounced products".

Apple's use of this veil is not always consistent. Apple often discusses future directions for Macintosh technology (such as for Mac System Software 7.0), but does not treat the Apple II equivalently; new features of GS/OS were not announced before the release of the operating system, for example. Apple is also not visibly aggressive in challenging statements such as those of the *Post* article mentioned when it benefits their "prestige" computer to remain silent. If Apple truly wants to quell rumors, they should apply their "standards" equally. But the approach of "watch what we do, not what we say" is laudable, if adhered to across all of Apple's product lines.

We hope Apple itself contacts publications that attempt to (negatively) influence sales of current Apple systems. Rumors such as those in *MacUser* are more puzzling, since they would appear to be more damaging to Mac sales than those of the Apple II. A \$995 Mac SE with 20 meg drive would probably greatly anger current Mac customers, who paid a retail price of over \$3500 for the same system.

Commercial Apple II publications (including ourselves) are not immune to repeating rumors; it's very hard to keep personal feelings out of the editorial task. But journalists need to be careful to distinguish what they report as news from opinions, and readers need to be analytical about what they read. Let's analyze a few of the rumors of this year, and do some qualified guessing about what it means working with what we currently know about Apple and its users.



"THESE KIDNAPPERS ARE CLEVER, LIEUTENANT. LOOK AT THIS RANSOM NOTE, THE WAY THEY GOT THE TEXT TO WRAP AROUND THE VICTIM'S PHOTOGRAPH. AND THE FONTS! THEY MUST BE CREATING THEIR OWN—MUST BE OVER 35 TYPEFACES HERE,..."

Consider first the perpetual rumor that Apple will (soon) release an "inexpensive color Macintosh" for the education market. Even a cursory knowledge of Apple Computer demonstrates that "inexpensive" and "Macintosh" never occur in the same sentence, unless we're referring to the Mac Plus (any Apple II Plus users out there want to speculate on the future of its Macintosh namesake?). The Mac Plus is still viable, but it seems obvious that the SE is the basic compact Mac platform of today, just as the Mac IIx is the basic compact modular system. Apple appears to consider the Mac to be a good value at current prices, and has been filling out the product line at the "high end".

Let's arbitrarily define "low cost" as a system retail price of under \$2000. The "inexpensive Mac" rumor's reality check is easily accomplished by reviewing Apple's current pricing policies for Mac video interfaces (\$499.00 for the basic 4-bit color video card) and color monitors (\$999.00 for the high-resolution color RGB monitor). Does it look credible that Apple will build a color Macintosh that will be cost competitive with Apple II or MS-DOS systems when the video display alone comes in at \$1500? This projects Apple building the most expensive entry-level system of the last decade, or (in the case of a dramatic cost reduction) devastating the revenue they've been making with the Mac in the business market.

Consider also the rumors of an inexpensive "dual processor" combining the Apple II and Macintosh. These rumors take the form of either the "Golden Gate" (basically a Mac with an Apple II coprocessor), a Mac running Apple II emulation software, or a "Mac coprocessor" for the Apple II. These all fail the reality check for various reasons.

An inexpensive Mac with an Apple II hardware coprocessor is unlikely for the same reasons stated for an inexpensive color Mac: by the time you add up the costs for the hardware to emulate both the Macintosh and the Apple II (with color), you're out of the mass-market consumer and education price range (Tom likes to refer to this rumored system as the Brooklyn Bridge). There is also the problem of finding a market for a product that probably would not be either the best Apple II or Mac it could be. Jean Louis Gasseé has repeatedly said that Apple does not build "Swiss Army Knives"; this seems to be a wise strategy. As recently as AppleFest San Francisco 1989, he also commented on the probable expense of such a system.

To emulate an Apple II in software at an acceptable speed using Computer Applications' *II in a Mac* requires a Mac with at least the speed of an SE-30, so let's place the minimum entry level for such a system at about \$4500 retail. Immediately, we're out of the "low cost" bracket again. Of course, you give up color and the ability to read 5.25 disks at that price. A IIgs emulation requiring greater speed, more hardware-specific features, and so on is right out of the question.

Apple II users, of course, clamor that a Mac coprocessor for an Apple II could be a solution. Again, the Mac and Apple II hardware is rather distinct; the cost of such a coprocessor would probably exceed that of buying a Macintosh Plus at current prices. Buying the separate computer means you don't have to tie up your Apple II while running a Mac program, or vice versa.

None of this is to say that Apple may not develop some computer in the future that merges features of the Apple II and Macintosh in a way that allows it to emulate either system. But if we look at the features and pricing of the two lines currently, they are rather distinct. Too distinct, reason tells us, to expect Apple to unveil a radically new "MacApple Swiss Army Knife" in the near future. Apple itself expects their theoretical Knowledge Navigator to be distinct from either current model.

Speculation should be tempered with a bit of logic, beginning with an analysis of what Apple needs to do to keep its lead in the educational market and gain a more secure foothold in the price-sensitive home market. Apple can claim "success" in the business market by making inroads against IBM, but they actually own a minority percentage after years of pushing the Mac as hard as they can (and using a lot of Apple II revenue to do it). Would it be reasonable to assume that, under increasing competition from IBM and Tandy, Apple's educational sales could decline to a minority of that market as well?

Let's move to the actual question: if educators currently using the Apple II want to have Apple's view of personal computing in an inexpensive package, then, how could Apple most reasonably provide it in an way that Apple can market effectively?

Mac proponents point to the success of the Mac in business ("success" is a relative term, of course), but education is a different market. Educational institutions do not expense equipment in the same way that businesses do; it can be assumed that initial cost and longevity of service will be a major consideration, as well as the ability to use existing equipment and software.

Much of this existing equipment will be Apple II related for many schools. Would you like to be the Apple salesman walking into a district meeting and telling the customer that bought 100 Apple II systems from you a year ago that those systems must be "retired" and replaced with new, incompatible systems? If you were the school district's representative, would you be dialing 1-800-GOMSDOS?

The next question: what is Apple's strength in technology? Apple believes strongly in their desktop interface metaphor. Apple's vision right now, based on what we saw at an Apple "Technology in Education" conference in Overland Park last spring, seems to be leaning heavily toward hypermedia and networking. Apple demonstrated these principles using Macintosh II systems costing upwards of \$5000 retail, of course; not exactly mass-market pricing, even with educational discounts. (It was interesting to notice that of the Apple IIgs systems present in Apple displays, none were running a desktop application. At Apple's booth at the 1989 Apple Fiesta, the IIc Plus they were using for demonstrations was not even set up properly; it was laid flat on a tabletop with no ventilation!)

The conclusion is that Apple needs a mass-market alternative, and this started the "cheap color Mac" rumors. But we've already discounted the existence of such a system in the near future. Or have we overlooked something? (Warning! Rumor approaching; engage your reality filters!)

This led me to a hypothesis that makes so much sense that I believe that Apple must be working on it, despite my doubt that Apple has enough respect for the Apple II to be developing such a product. I hope the latter feeling is the one that's wrong; if Apple is working on this project, it will blow the doors off the creeping intrusion of MS-DOS machines into our schools, provide the less myopic of the Mac devotees with an alternative they can live with for the future of personal computing, and transmute Apple's image among Apple IIgs users from lead to gold.

Apple obviously has a lead in the use of hypermedia, primarily launched by the inclusion of *HyperCard* with Macintosh systems. But as you start adding features to the Macintosh to support what hypermedia involves (interactive audio and video, and so on), the price tag ends up out of range of the average user. With that \$5000 Mac II at the center of the AppleCD SCSI drive, Pioneer *LaserVision* player and Sony monitor employed for demonstrations, hypermedia is in a bracket most schools can't justify as an expense except in specialized circumstances. Apple has prided itself as "the personal computer company"; obviously, not too many persons can afford a \$6000+ investment.

If we go with a bare-bones "HyperCard engine" such as a Mac SE, we're only talking something over \$3000 retail for the computer. Educational institutions get a discount, of course, but many elements of Mac expandability are expensive. Look for an audio digitizer, for example, and you find Farallon's *MacRecorder* at \$295. Assume the school is willing to trade off some ease of use and capability for fiscal survivability, and you see MS-DOS emblazoned on the horizon.

If you're Apple, you don't want to let the MS-DOS faction get a foot in the door; ideally, you don't want it to be considered a viable option. You have to replace the "HyperCard" engine with something more affordable to educational institutions; that "cheap color Mac" is needed in a hurry. But you can't afford to get your current business customers angry with you by bringing out hardware with the approximation of Mac II features in a price range that undercuts their investment severely. You also have to avoid walking into an education institution that has invested heavily in the Apple II and asking them to throw it all away and convert to Macintosh systems and peripherals. Apple is between a rock and a hard place. Unless they use the other Apple computer that already supports the desktop graphical user interface (in color, no less) and some of the other necessary peripherals: the IIgs.

Now we accept (necessarily without proof) that some recent rumors about Apple working on a IIgs version of *HyperCard* are true. Assume that Apple will extend the compatibility of the two lines by making the IIgs *HyperCard* data structure the same as the Macintosh version where possible. Further assume that part of such a product will be a IIgs GS/OS FST to allow reading the Mac stacks directly from Mac disks; in this case we have the advantage of knowing that Apple has designed the capability of adding such an FST into GS/OS. We also have to assume that, as for the Macintosh version of *HyperCard*, the product will require a minimum of one megabyte of IIgs RAM memory.

What would you call a color computer system that could execute Mac *HyperCard* stacks from a Mac disk?

Apple will have wrought a new computer in a software package! No revolutionary new hardware for the IIgs is required. Every current user of a IIgs with at least one megabyte of memory, an Apple 3.5 drive (to read Mac

disks via the FST), and (if desired) a color monitor will have the new Educational Color IIGs (some misinformed people spell that "Mac") for the price of a box labelled (for want of a better name, for now) "HyperCard GS".

The cost of upgrading for CD-ROM becomes only slightly higher than for the Mac (the IIGs requires an Apple II SCSI Interface card; the Mac's SCSI interface is built in). The cost of adding a color monitor itself is much less; the IIGs monitor is less expensive and requires no expensive interface. The cost of adding the video disk player is the same; unlike the Mac, where you would need to buy a separate monitor for the videodisk player, you can elect to spend the money on an Apple II Video Overlay Card for the IIGs and use the IIGs monitor. Notice the latter option allows you to overlay video and computer displays for enhanced effects! Apple cuts the price of the hypermedia engine in half, possibly at the loss of some capabilities, but with the addition of some others.

Where does this leave *HyperStudio*? Probably with superior features; Roger Wagner himself says that *HyperStudio* was not designed as a clone of *HyperCard* for the Mac, which we have postulated is the tac an Apple *HyperCard GS* must take. As mentioned, a representative Macintosh audio digitizer is \$295. On a IIGs, you can buy *HyperStudio* with a digitizer included for less than half that, and have an enhanced hypermedia product to boot. *HyperStudio* will probably remain the premiere IIGs hypermedia tool for schools wanting to specialize in applications taking full advantage of the IIGs. *HyperCard GS* would primarily be of interest to educators who are using Macintosh and IIGs computers in a shared environment; it would allow them to cross-develop stacks, and to add Macintoshes to the environment on a schedule they can live with according to their goals and fiscal abilities.

Apple must be sensitive to the increasingly vocal Apple II owners asking how the company intends to support them. The impact of a *HyperCard GS* upon Apple's credibility among current IIGs owners would be profound. In the wink of an eye, many of the thousands of *HyperCard* "applications" attributed to the Mac become IIGs applications, too. Apple immediately lends respectability to the IIGs. Could even low-priced MS-DOS or Amiga systems compete against a properly marketed IIGs that runs current Apple II software, newer IIGs software, and can utilize data files for the most prevalent application for the Macintosh?

Apple would, of course, have to follow this by encouraging other serious applications for the IIGs; here we can't honestly say that we have seen evidence of Apple doing this. One test will be whether Apple corrects the balance of evangelism and marketing efforts on behalf of the Apple II; education must reflect the real world, and Apple must quit directing all real-world applications to the Macintosh. Let the Mac be the high-end machine it was designed (and priced) to be, and the IIGs can be the embodiment of the desktop metaphor for the rest of us (who don't have 6 figure incomes).

Do any current facts run in conflict to this scenario? The reason for the new "slightly revised" IIGs would become apparent; the larger amount of minimum memory has made the "base" IIGs into a machine capable of running this vital new program. If and when a faster IIGs arrives, its advantages will be perceived, but the current IIGs becomes the machine that revives the Apple II's marketability. Older machines with the added memory share the wealth.

It would also make AppleShare more attractive. Mac and Apple IIGs users would have an important application in common that can use the same data files. Just put the file in a public location on the server and let the users share.

It would also kick the various Mac publication's rumor mills in their collective teeth; Apple's admonition for Apple II users to "watch what we do" would be grandly justified in one tremendous step. After months of Apple II bashing and taunting, those conducting anti-Apple II propaganda campaigns, imagine facing a new Color "Mac" in IIGs clothing. Other than the nameplate, the functionality of the system for *HyperCard* (other than running Mac-dependent code) could be the same. The MacKrishnas will be left to explain why a computer that can operate like a Mac is a "toy". And they'll finally look as bigoted as they are.

If Apple is this smart (the thought "devious" comes to mind, in a good context), many of us should send Mssr's Sculley and Gassie a bottle of their favorite libation in place of the scornful confrontations they've had to endure this last year. Only time will tell.

In the meantime, don't take the rumors (including this one) to heart without qualification. Quoting from *The Abyss* (Orson Scott Card wrote the novel based on James Cameron's screenplay), "You have to see with better eyes than that."-DJJ

Miscellanea

Apple will be conducting several holiday promotions through the end of 1989 that include the IIGs.

The first is an option for a "Free Trial Run or Cash Back" on a qualifying Apple IIGs or Macintosh system from October 14 through December 31, 1989. Customers that meet Apple financing criteria and purchase a IIGs system with Apple monitor and disk drive can get \$150 cash back on an ImageWriter II or LQ printer, LaserWriter IIGS, IINT, or IINTX printer, external Hard Disk 20SC, AppleCD SC CD-ROM disc drive, Apple Scanner, or Apple Personal Modem (notice that IIGs-specific drivers for the ImageWriter LQ, LaserWriter IIGS, and Apple Scanner aren't available at this time; they appear to be there for Mac participants). Or the customer can shun the rebate for a peripheral and instead take the system home for a Free Trial Run, to be returned without cost between January 1 through 15, 1990, at no cost if the customer elects not to purchase the system. If a decision to purchase is made before December 31, 1989, and the purchase is not part of an Apple Operating Lease program, a \$100 bonus is given off the price of the system.

Several companies are also offering money-back offers on software titles in conjunction with the promotion. The software can be purchased or returned directly to its manufacturer if the Apple system is not kept; if the system is kept, the purchased software must be kept.

Contact your Authorized Apple Dealer for further details.

Apple IIGs System Software 5.0.2 has been released; it fixes many reported bugs including the BASIC.SYSTEM bug mentioned in the September 1989 issue of *A2-Central*. The software should be available from Apple dealers as well as from information services with licenses from Apple for electronic distribution (the version Apple provides for electronic distribution thoughtfully includes a smaller "update" disk for those who already have 5.0; this will save download time and dollars). We'll also have an update on this month's disk.

The timely release of the revised system software indicates the commitment of the Apple II teams at Apple. It may also indicate an awareness of the Apple II reviving among the administration at Apple.

Meanwhile, Apple Direct delivers a slap in the face from Apple's other hand. In an article titled "And the Survey Says..." by Lisa Raleigh (pages 13-14), noting "a relatively high percentage of responses from Apple II developers", we find a report of a count of 6598 Macintosh developers, 1019 Apple II developers, and 1769 developers for both computers. Then, in the best tradition of disinformation, the paragraph "interpreting" the numbers says:

The survey had a proportionally high response from Apple II developers (although many of the developers who said they were primarily Apple II-oriented may fall into the "Both" category).

After several years of experience as a chemist, I have developed a great disdain for individuals that accrue data and then discard information that doesn't suit their purpose. Either the respondents are Apple II developers, or they aren't; I don't see a justification for why they "may fall into" another category. I also don't see a similar condescending statement that Mac respondents "may fall into" the combined category.

The following sentence in the article notes that:

Perhaps Apple II developers are especially motivated to respond to surveys like this!

It would seem to be a good survival tactic, since Apple pays little attention to their numbers. Apple makes life very difficult for Apple II developers; having reduced their Evangelism and Marketing commitment to the Apple II by constraining the Apple II's "target market", siphoning money from this induced loss of profitability by charging \$500-\$700 a year (the same as Mac developers) for the developer's programs, then using that money to print drivel like this. Is it any wonder Apple II developers want an independent association?

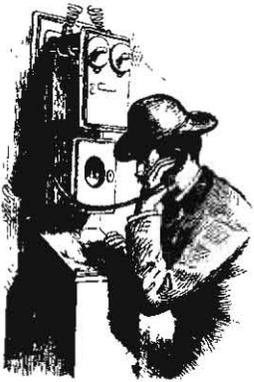
Laser Computer is taking advantage of Apple's callousness. Laser is offering to place a coupon pack made up of discount-offering ads from various developers in the boxes of the 10,000+ Laser 128 Systems they ship each month. Interested developers should contact Laser at (312) 540-8086, extension 743 or 744, for further details.

SoftView, the company that produces the TaxView program for Macintosh and MS-DOS systems, also sold a version for the IIGs last year. SoftView has sent a letter to IIGs owners indicating the program has been dropped "due to decreasing sales of the IIGs". As far as we know, decreas-

ing sales are news to Apple; this will be a test for Evangelism, who should at least correct any mistaken impression on SoftView's part, and (if they are mistaken) ask that SoftView not make misleading and potentially damaging comments regarding the Apple II market.

If you want to use AppleWorks to do your taxes, 1040Works is a package of AppleWorks templates, macros, and other utilities that can compute Form 1040, Schedules A, B, C, D, E, F, R, and AE, and Forms 2106, 2441, 4562, 6251, 8582, 8606, 8615, and 8914 along with their worksheets. 1040Works has been updated for AppleWorks 3.0, runs on a 128K machine, and costs \$24.95. 1040Works-X includes some enhancements but requires at least a 90K (expanded) desktop, it costs \$27.95 (both prices are for 5.25 disks; for 3.5 versions, add \$3). Both are available from Personal Financial Services, P. O. Box 1401, Melville, N. Y. 11747.

Claris announces a long-awaited upgrade to AppleWorks GS. Version 1.1 addresses several complaints, including speed of loading the modules, speed of printing, compatibility with IIgs System Software 5.0 and the new ROM 03 IIgs. Draft mode printing has also been enhanced, AppleWorks 3.0 files can now be imported, and custom dictionaries for the spelling checker may now be larger. The suggested retail remains at \$299; registered AppleWorks 1.0 users can obtain the new version for \$29 (those who purchased AppleWorks GS after September 15, 1989, will be upgraded at no charge). For upgrade information, call Claris at (800) 544-8554.



Ask (or tell) Uncle DOS

nizes the first two. I much prefer to use a number of REMark statements at the top of my program listings that explain the use of all variables. It has helped me immensely when I later "improved" them. It also allowed others to modify my programs for changed conditions (I have used this technique when programming in languages that recognize more letters because it is a big help to have all variables defined in one place.)

You also use subroutines for code that is only used once. This only complicates the listing and makes it harder to follow. Code that is used only once should be put directly where it is used.

In your answer to John Rohan's letter (5.78c) you did a good job of what OPEN does but you beat around the bush on what READ and WRITE do. Why not say "READ redirects all INPUT and GET statements so they obtain their input from the specified OPENed file until the next PRINT Control-D is encountered. In like manner, WRITE directs all PRINT statements to the file until the next Control-D. Note line 1260 in the BASIC database redirects input to the keyboard.

As an aside, line 1320 of the BASIC database is simpler if written:

```
1320 IF LEN (CMD$) >1 THEN R = VAL
      (MID$(CMD$,2)).
```

The result is the same since, without the third operand, Applesoft uses all of the string from the second operand to the end of the string.

J.E. Douglas
Baton Rouge, La.

The explanation of OPEN could have contained more detail on the importance of control-D. Your explanation (and the example using line 1260 of the database program) makes a good point.

I prefer to try and use "mnemonic" variable names. Applesoft programmers should recognize that only the first two characters are significant in differentiating the names for each variable type; at least, we haven't received a letter claiming confusion (yet). A more fully self-documenting version of the program could include a full description of the variable use. The line between describing the particular program presented and teaching Applesoft all over again has to be drawn; we depend on comments like yours to keep us "honest".

The "complete" version of HyperStudio is shipping to registered users who have bought the previous versions. If you haven't registered your copy with Roger Wagner Publishing, send in that warranty card!

We believe HyperStudio may be the program that "makes" the IIgs, as VisiCalc did for the Apple II, AppleWorks did for the IIe/IIc, Lotus 1-2-3 did for the IBM PC, and Pagemaker (and other desktop publishing programs) did for the Mac. (See this month's catalog for an important announcement.)

Applied Engineering has a new 300/1200/2400 modem called the DataLink Express. It's an external model that has a unique Line Engage indicator to tell you if the phone line the modem is connected to is in use, a very handy item if you're using the modem on an extension. The modem includes telecommunications software for Apple II, Macintosh, and MS-DOS systems, has a five-year warranty, and retails for \$249. Options include MNP-4 error correction (\$89) and a send-only FAX chip (not yet available).

How do you repair a Macintosh drive? American Educational Services, 7611 Allman Drive, Annandale, Vir. 22003, (703) 256-5315 manufactures a Drive Exerciser for \$269 that allows you to use a Dysan Analog Alignment Diskette (also available at \$50) to test and align Mac 400K and 800K 3.5 drives as well as Apple II 5.25 drives. Of course, we wouldn't be mentioning it here unless it worked on an Apple II; the listed requirements are a IIe with a 5.25 or 3.5 drive, an oscilloscope, and the alignment diskette.-DJJ

The use of RIGHT\$(CMD\$,LEN(CMD\$-1)) rather than MID\$ is also personal preference; I read it as "return the right part of the string excepting the first character". Your MID\$ example is more compact (and possibly faster to execute), but it seems less explicit to me.

I'd have to disagree about the structure of the database program in general though this certainly falls into subjective preferences for the style of a program.

I attempt to modularize a program as much as possible because it follows the way I plan a program (and presented this specific program); I develop the overall scheme of operation, construct the minimal skeleton, and then fill in the details. Such top-down design isn't very critical for a short program like the database example, but it definitely makes the program's structure and operation easier (for me) to interpret when I come back weeks (or months) later to make revisions. If you look closely, each of the subroutines is executed only once each time it is called, but it can be called many times as the "command interpreter" (in lines 1280-1460) is repeated. Attempting to do the same thing in linear coding would result in GOTOs branching throughout the program: the dreaded "spaghetti code". GOTOs should be reserved for simulating conditional branching operations that Applesoft doesn't inherently support through its keywords.

The use of subroutines was also intended to allow the various modules of the program to be revised. Each module can be treated as its own "subprogram", and this limits the scope of the code that is directly involved in changes. There are some limits to this independence in Applesoft; for example, there are no local variables so that the effect of a subroutine changing the value of a variable on the rest of the program has to be considered (and this would be a good reason to document the use of variables in REM statements within each subroutine). Actually, the program was written with the idea that it eventually would not be done in Applesoft; IIgs owners may want to take advantage of their extended memory and code the program in decidedly modular languages such as C or Pascal.

For some reasons to revise the program, we can look to a professional who does these things for a living...-DJJ

Corrections and amplifications

The database program in the last issue is a little too sincere about letting you abort "P"acking the database. If you select to "P"ack and then answer "No" to the "Is this what you want? (Y/N): " prompt, the program will crash with a BASIC.SYSTEM "FILE BUSY" error. The problem is the program branches to the routine to re-open the files used by the program whether or not packing was performed. We can fix this by changing line 1360 to read:

```
1360 IF C$ = "P" OR C$ = "p" THEN GOSUB 1920:
      ON P + 1 GOTO 1290,1160: REM pack
```

to let us select where we should go based on the value of P, and then adding the line:

```
1955 P = 0: REM no pack yet
```

to signify we haven't packed the file (yet), and:

```
2035 P = 1: REM now packed
```

to indicate the file has been packed.

Triad Software is now advertising their Ultra.Invoice program (mentioned in the September A2-Central) at \$75. Triad also plans to offer a separate inventory program in December 1989 for \$75.

BASIC style

I feel that you have done a disservice to inexperienced Applesoft programmers on at least two counts and could have been more helpful on a third.

In your BASIC database, you use variable names of more than two letters without any mention of the fact that Applesoft only recog-

Too quick and dirty?

Having read your "Quick and Dirty Database" article, I thought your readers might be interested in a few comments on ISAM systems and a new product that fits almost all of the criteria you are looking for.

ISAM ("indexed sequential access method") has almost as many definitions as there are database management systems that claim to use it. The system you described has several disadvantages. Among them:

- The records are not stored in a useful order—they must be sorted for any meaningful reporting functions other than the order in which they were entered.
- While your search by record number is fast, searching for any other criteria (last name, client number, part number, etc.) will be very slow.
- If an edited record is larger than its original length, it must be added to the end of the list, leaving another inactive record.
- The index file grows proportionally with the main file. At nine bytes per record, the index for your test file of 3,430 records would be over 30K - already too large to fit in RAM with your program. A file of 50,000 records would require an index of nearly half a megabyte!

Here's how the "ISAM" system in *DB Master works* (keep in mind that most of this is totally invisible to the *DB Master* user):

The first field (or fields) in each record are defined as the "primary key".

Records are stored in "data blocks". Each data block can hold from one to as many variable-length records as will fit. In our case, the size of a data block is 4K, so they typically hold up to 30, 40 or more records. The records within each data block are stored in primary key order.

There is an "index" that includes the highest key in each data block, and a pointer to where that data block can be found within the file. Note that that reduces the number of records that must be indexed by a factor equal to the average number of records per data block.

To locate a record by its primary key, the program merely has to do a quick search of the index entries (which are in RAM) to find out which data block holds the record. It then loads that data block, and does another quick in-RAM search to find the record.

That's fine as far as it goes, but the index for a 50,000 record file is still going to be too large for a 128K computer. So we add a second level of indexing: we divide the file into "volumes", each with its own "volume index", and we add a "master index" which includes the highest key in each volume. Now to do a primary key search the program has to:

- search the master index (in our case, never more than 50 entries) in RAM to find out which volume holds the record
- load that volume's index into RAM (unless, of course, it's already there)
- search the volume index (again, no more than 50 entries) to find which data block the record is in
- load the 4K data block (assuming we don't have it already)
- find the record

To find the next or previous record in the pri-

mary key order, you don't even have to refer back to the index.

Now, that may all sound a bit complicated (it is), but look what we've accomplished: We can index an entire additional 200K volume for just a few bytes (the key length plus some overhead). And we can index a 10 megabyte file (the number of records doesn't matter), and never need more than about 5K of RAM for indexes, even with a 50 byte primary key!

And of course this is all written in machine language, so it's fast. Note that the total difference between searching for the first and last records in a maximum size file (by their primary keys) is a few compares in looking at the master index - a tiny fraction of a second.

To add a record, the program figures out where it belongs, loads that block, and inserts the record.

To delete a record, the program merely closes up the hole in the data block, and, if it was the highest record in the block, adjusts the index.

Records can grow or shrink as needed. If a record won't fit in the data block where it belongs, you simply split the block in two, and adjust the index accordingly. (Well, OK, it's not really that simple, but I'm not about to give away all of our secrets!)

As you can see, there's quite a difference between ISAM systems.

Meanwhile, if anyone has bothered to read this far, you may be interested in the new product that should be available by the time you read this. Once you've created a file with *DB Master*, the *Basic Programmer's Pak* will let you search, add, edit and delete records in that file from your own Applesoft Basic programs. With the necessary binary files, indexes, etc. loaded, there is room over 25K of Basic code and variables. All filing operations are done via a few simple ampersand calls, and the program communicates with your code using normal Applesoft variables.

With two exceptions, the combination of *DB Master* and the *Basic Programmer's Pak* meets all of your goals: files can grow up to 10 megabytes, data is stored in a highly compacted form, the full ASCII character set is supported, access is very fast, and the commands and data format are as "extensible" as your Basic skills can make them. The data files are not, however, stored in a standard text format, and the Pak cannot create its own files - that requires *DB Master*.

Barney Stone
Maple Glen, Penn.

The system I described for the sample program would hardly be called ISAM by real-world programmers. Obviously, a "hidden criteria" is that the article describing the technique had to fit in a single issue, with all of the limitations Barney described in explaining the advantages of his carefully designed commercial program. Hashing operations are a subject unto themselves.

The Basic Programmer's Pak is available from Stone Edge Technologies, Inc., P. O. Box 3200, Maple Glen, Penn. 19002. It sells for \$129.-DJJ

Forms generation

The irony of this note is that it is being written on an IBM (at work) about a good machine...

I've got a friend who has bought a IIc-Plus;

now she wants to use it in her real estate business. She's ordered AppleWorks 3.0, but there are many specialized forms that the real estate business uses: forms for HUD and the like. She showed me a catalog which lists several IBM programs to print these forms. None were listed for the Apple. If you don't know of programs like this (not property management, but sales and appraisal forms) could you ask your readers for help? You may publish my address in **A2-Central** if you wish.

Commentary: Some time ago you had a few articles on CP/M and Logo. I'd like to see more of that type of thing. I don't know enough to contribute to them, but I'd like to learn more - not just beginner information. For example, I have *Ace Calc*, which is the same as *Magicalc*. It uses the 64K on my CP/M card without a pre-boot. How? I'd like to know how to do that in my own programs and, perhaps, how I could adapt unprotected commercial programs to do the same. My copy of AppleWorks 2.0 doesn't do that; could I modify it to do it?

Keep up the good work. If mailing rates were based on information content instead of weight you couldn't afford to be in business.

Willard Seehorn
Columbus Computer Services
P. O. Box 602
Whiteville, N.C. 28472

Forms generation is certainly something the Apple is capable of, but we've not seen a great many of them on the market.

For AppleWorks, the solution would probably be to create a word-processor document equivalent to the form and use the built-in mail-merge functions to fill the form from a database. If AppleWorks' mail-merge can't keep up, you may need to go to something like TimeOut ReportWriter, or to a database program (have we mentioned DB Master recently?) that has more detailed forms generation.

The use of AppleWorks does assume that the form is laid out in the manner of a typewritten page, and that AppleWorks can either satisfactorily duplicate the form or fill in a preprinted version of the form. A product called GRIDS+ prints lined forms, grids, and other forms designs using SuperFonts.

AceCalc and its other variations contain routines to check for various types of memory expansion cards; there's no "magic" in the use of the memory on your CP/M card other than the program contains the necessary information to know how to access the card. There was a discussion of RAM card protocols in "A concise look at Apple II RAM" in the December 1986 issue; if your CP/M card RAM doesn't conform to one of these then you'll have to see if the manual defines the access method or if someone else knows...and can tell; sometimes such information is only revealed to those that sign non-disclosure agreements.

The PlusWorks program may allow AppleWorks to recognize the RAM on your CP/M card; contact Norwich Data Services, Ltd., P. O. Box 356, East Norwich, N.Y. 11732, (201) 679-0594, to see if your card is supported.-DJJ

AppleWorks and the II-Plus

In your July issue, you discuss AppleWorks 3.0, but neglect to mention whether or not it will run on a II-Plus. I'm presently using v2.0 that has been modified with *PlusWorks* to run on my Franklin 1000. Will there be a comparable program from *PlusWorks* to modify v3.0?

Even though I can run v2.0 on my machine, and the program takes advantage of my 1 meg RamFactor (Applied Engineering) card, none of the macro programs will run on my machine. All they will say is something like "requires modified lle". Do you know of any way of getting Beagle Bros' macros to run on my set-up? (Or comparable programs?)

In your August issue (p. 5.54), you mention patches for AppleWorks, in particular John Link's *SuperPatch* program. But you say "If you want a patch for AppleWorks 1.x or a patch to make AppleWorks running on a II-Plus do something, we don't know where to send you".

I just want to mention that I used Link's *SuperPatch 3.1* to modify AppleWorks by going back to the original, unmodified version of the program. I ran *SuperPatch* first, and made the changes. Then I ran *PlusWorks* to configure the program for my system. It works fine, and I'm looking forward to trying v5.0 of *SuperPatch*. John, by the way, was quite helpful when I had some initial problems. It turns out I was using an old version of BASIC.SYSTEM. When I installed the later version (v1.2; 14 Dec 87), *SuperPatch* worked perfectly.

Michael Wogan, Ph.D.
West Collingswood, N.J.

AppleWorks 3.0 will require an "expander" for the Apple II-Plus, as for earlier versions. With AppleWorks 3.0 only out a few weeks, it may take a while for one to appear.

The macro programs are probably looking for an enhanced lle at a minimum; whether the program can be patched (in a manner similar to AppleWorks) to work on an Apple II-Plus is unknown to us. Randy Brandt (author of TimeOut UltraMacros) has indicated to us that the program assumes the memory configuration of a 128K Apple IIe or equivalent; Apple II memory expansion cards do not mimic a 128K IIe.

Thanks for the information on the suitability of SuperPatch for the II-Plus. The problems you were having with BASIC.SYSTEM may relate to the difference in the way v1.0 and later versions handled the BSAVE command.-DJJ

Fixing a date

I believe I have another solution to Keith J. Scala's problem with truncating of 1989 to 89. When I use the database for dates that need to be written out, I use a separate column for the year. Then I "Arrange chronologically" on the Date column, followed by "Arrange 0-9" in the year column.

My print layout takes care of the correct right justification spacing by way of open-apple-J.

Date	Yr
Feb 29	1960
Mar 15	1989
Apr 21	1988
May 26	1978

<- this is how data is first entered

Date	Yr
Feb 29	1960
May 26	1978
Apr 21	1988
Mar 15	1989

<- finished product

(printed)

Liz Anderson
Mt. Holly, N. J.

Periodically I make an AppleWorks database file from a mainframe-generated text file. The file consists of personnel data of about 900 individuals. I copy the records from this file to the clipboard and move them from the clipboard into another AppleWorks database file which has the correct category names, report formats, etc. I've written a macro to convert entries in date fields to a format that AppleWorks can sort chronologically.

I've discovered that if I copy the records to the final ADB file rather than move them AppleWorks 3.0 automatically converts entries going into date fields to the appropriate format. I don't believe this feature is mentioned in the manual.

I keep discovering little features like this that suggest that the modifications to version 3.0 were made by people who understand AppleWorks and the people who use it.

Leigh S. Rockwell
Wyomissing Hills, Penn.

When the chips are down

Whenever I booted into Prodos 16, I would get the error message "Unclaimed Sound Interrupt 08FF" and the bouncing apple behaved very erratically. It looked sorta spastic, moving part way to the left and then restarting. Prodos 8 would boot just fine, except when I went into the control panel, the machine would die at random intervals in the control panel. Sometimes it would quit at the main panel, giving the error message as above, and leaving the time and date (!) on screen. Other times it would die in the sub sections, for example, in the RAM disk settings panel. No rhyme or reason, it just quit.

The fix: remove the Enqsonic chip (the large one on the lower left by the RAM card connection). It helps to remove the trim strip with the power on light. (That is done by gently bending the locking pins on the bottom of the case) After the chip is out, carefully clean the pins with an ink eraser. Reset the chip. I have had no problems since I did this.

Jim Coffey
Coram, N.Y.

If you fear the eraser is too abrasive, you can try a commercial "contact cleaner" sold at electronics stores.-DJJ

TDD information

In the October 1989 *A2-Central* (p. 5.69), James Ketchum of Kalamazoo, Mich. asked about communicating between Telephone Devices for the Deaf (TDDs) and Apple IIs. While not an expert in the area (of either), I may be able to offer some assistance.

As I understand, TDDs normally communicate in Baudot format, a five bit code. Since Apple IIs use ASCII format instead of Baudot, a direct communication between Apple IIs and TDDs would not be possible. Mr. Ketchum's only hope, as I see it, is that the individual with whom he is communicating possesses a TDD that incorporates an "ASCII key". Such a key enables the TDD user to communicate with almost any computer system, assuming it possesses a modem and the appropriate communications software. If he, or other *A2-Central* readers are interested, this and other information is available in an article by Michael Pflaum entitled "The California Connection: Interfacing a Telecommunication Device for the Deaf (TDD) and an Apple Computer". This article appeared in the September 1982 edition of *The Ameri-*

can Annals of the Deaf (pages 573-584).

I hope that this answers some of Mr. Ketchum's questions and provides an additional resource to those interested.

E. W. Testut, Ph.D.
Associate Professor of Audiology
Ithaca College
Ithaca, N. Y.

In reference to James Ketchum's request for TDD information, I have something that may help. TDD's communicate at 45.5 baud and use Baudot code. This is a five-bit code.

Instead of going to all the trouble to modify his modem and write his own communications software, he could try to find an old Novation *Apple Cat II*. This modem could be modified (one jumper, done free at the factory) to be used as a TDD. The software to support this was called *Deaf-Ware* and was sold by Novation for about \$30.

Until being sent to Germany, I used my *Apple Cat* for a TDD terminal for six years, first in a II-Plus, then a IIe, and now my IIGs. There shouldn't be any compatibility problems with whatever machine he has. I have never seen another modem that could be used for TDD. Hope this helps.

Keith W. Higgins
Landstuhl, West Germany

Quantum hard disk formatting

I have read in one of the GENie BBS's that Mac hard drives using the Quantum mechanisms are formatted with a 1:1 interleave which can't be altered. I have a IIGs and am considering a MAC HD (40 MB). What effect will this have on a IIGs's performance (i.e. access speed, etc.)?

John T. Chiao
Hughesville, Penn.

Chris Adams of Chinook Technology points out that the Quantum Pro series drives (40, 80, and 105 MB capacities) have a built-in cache that should remove any influence of the interleave on the access speed.

Judging the best interleave for your disk drive is complicated; it depends on many factors down to the software you use most often. Unless you're anxious to spend quite a bit of time testing boot, save, and load times with your usual software with various possible interleaves, consider if the speed of the "standard" formatting is adequate for you to live with.-DJJ

Megaboard found

In reply to Mr. Thomas J. Scott's letter published in the November issue of *A2-Central*, the manufacturer of the *Megaboard* is:

Prisma Datensysteme GmbH
Sofien Strabe 30
6839 Oberhausen-Rheinhausen
WEST GERMANY
The U.S.A. distributor is now:
DMA Technologies, Inc.
601 Pine Avenue
Goleta, Calif. 93117
(805) 964-0733
(805) 964-0794 FAX

When you buy this product, you must specify the Apple IIe or IIGs version, as there is a difference in one of the PAL's. DMA supplied the necessary software with the IIGs version and should be able to solve Mr. Scott's problem. I use a *Megaboard* on my IIGs with a Seagate ST-125 20 megabyte drive (28 ms.) and a separate power

supply. In Brazil there is no after-purchase service for Apple hard drives (*InSider*, *Vulcan*, etc.), so the *Megaboard* was my best option and I'm very satisfied with it.

David James Rotenberg
San Paulo, Brazil

Another (not so bloody) virus

If you've experienced strange color changes of the border after releasing open-apple-control-reset or color changes of the background and the border, you've got a virus in your system.

The virus will delete blocks 0 and 1 on 3.5 disks in October. Furthermore the screen will turn read and the following message will be displayed:

```
System failure in +++
08
```

The number means seconds and is counted down. Reaching 00, the screen will turn green with the border cycling through the colors and the following message will appear on your screen:

```
0013 Copies Distr: Artistes Associes
== LOAD RUNNER ==
```

```
Premier Virus NON-DESTRUCTEUR su IIGS
```

```
par SUPER HACKER & SHYRKAN
```

```
du MASTERS CRACKING SERVICE 1988 Lyon
```

The virus copies blocks 0 and 1 of 3.5 disks and can be identified through the bytes \$01 A9 50 at the beginning of block. Block 1 contains the code for the color changing. An uncorrupted block 0 begins with the bytes \$01 38 B0. In this case block 1 is either empty (containing all 00's) or has reference to the SOS operating system of the Apple III.

The virus installs in RAM and can be deleted only by shutting down the system power or by initiating the built-in self-test routines of the IIgs. If in RAM the virus installs itself on every unaffected 3.5 disk in the drive with every open-apple-control-reset. It seems it installs only on the disks in the boot drive. I have some data disks which I usually put into the second 3.5 drive which were not affected, although the program disk was.

To eliminate the virus from the disks is rather simple. Just use a sector editor (like Copy II Plus) to overwrite blocks 0 and 1 with the same blocks from an unaffected disk.

Udo Huft
Wittmar, West Germany

The virus does seem to be relatively innocuous; overwriting blocks 0 and 1 but leaving the rest of the disk unaffected. When it "goes off" it forces you to boot from an uninfected diskette and restore the boot blocks of all infected disks; it's an irritant more than anything else. We've included a fix on this month's disk which was provided by Brian McCaig, who was one of the first to report the virus in the U.S.

To explain the difference in the contents of block 1: originally, the boot code blocks 0 and 1 contained code to allow the disk to boot on an Apple II using ProDOS or an Apple III using SOS. When GS/OS arrived, the boot code was modified. -DJJ

GS/OS and third party software

Is there a way in which to use GS/OS with programs like *Rocket Ranger*? It seems that the tools on the /System.Disk work fine, including

the System.Setup directory, but the program seems to be incompatible with something else. It boots fine but stalls after displaying the first graphics screen.

Also, what is the use of the FST's directory and which ones in this are essential for booting the computer?

Shantul Nigam
Southington, Conn.

*The size of GS/OS (with all associated files) may preclude installing it on a floppy with another large program. Some vendors may elect to include GS/OS on its own disk (as was done with **AppleWorks GS**); some may eventually elect to supply just the program, leaving the user to start up his computer with his own system disks.*

Among the Installer scripts for 5.0 is an "Latest Sys. Files (No Finder)" option; such an installation takes about 641K of disk space. If you can make a backup of your program disk (some may be copy protected, which precludes this), you can try this installation on a copy of the program disk to see if it works. If not, you'll have to do the disk shuffle unless you have multiple drives or a hard disk.

We've noticed programs that do not work properly with later versions of GS/OS, or with certain environments such as AppleShare. It's important to let the manufacturer know about these compatibility problems so they can attempt to solve them. Manufacturers that complain about "poor IIgs sales" should evaluate whether compatibility problems of their own doing are contributing to some consumer resistance. Programs that don't run with AppleShare and current system software versions get shelved fairly quickly here; this includes products that use copy protection schemes that preclude hard disk or network installation (for a single user, within the product's terms of use).

The FST's (file system translators) are GS/OS's method of communicating with various devices in a standard way. You need to have the FST's for the file system(s) you use (normally ProDOS; possibly AppleShare or others) and for the devices that may need support (CHAR.FST for character devices, such as the keyboard). Normally, Installer should install the ones you need. -DJJ

Joy(stick) with a mouse

You might advise Mr. Pagani (5.38a) that Happ Electronics, 3850 N. Main Street (or 4640 Island View), Oshkosh, Wisc. 54901, (414) 231-5128, make several "socket extenders" for Apple II Plus, IIe and IIc models. These include ones that allow more than one joystick or mouse to be attached to the IIc at once. While typically sold to allow more than one joystick to be active at once for multi-player games, they would probably work for what he wants. Contact Rick Happ, president of the company, for more information and current pricing.

I have been using one of the extenders they make for the IIe/IIgs for a couple of years now with no problems, and my experience with support for other programs they sell has been excellent. I recommend them highly.

Jack Countryman
Greenburg, Ind.

In response to the question posed by Mr. A. Pagani in the November 1989 issue, page 5.78, "Joyful mouse", an additional solution is the *Moustik Switch*. This small, solid-state device

plugs into the joystick/mouse connector on the back of the IIc and provides two separate connectors into which a joystick and mouse can then be inserted simultaneously. Pressing either the "fire button" or "mouse button" allows easy alternate selection of either device. My *Moustik Switch* has worked flawlessly since purchasing it over two and one-half years ago. The price at that time was \$29.95. The switch is manufactured and sold by Southern California Research Group, P. O. Box 593-R, Moorpark, Calif. 93020, (805) 529-2082. I recently called to verify their continued existence and production of the switch.

Roger A. Maltz, M.D.
Evanston, Ill.

Font Conversions

Are there any programs for converting Macintosh fonts into Apple II fonts? I have a unique font I'd like to convert!

Robert Kozlowski
Concord, Maine

The font file format is described in the January 1989 issue ("Fonts: on screen and in print").

A2-Central Font Librarian Mark Collins can convert the font for you and assign it a number, if you deign to send him a copy. You can reach Mark at:

Pebble Valley Software
2511 Pebble Valley Road
Naukesha, Wisc. 53188

*When the IIgs was introduced, Apple Computer sent some developers a Mac utility called **Font Munger** that could be used to prepare fonts to be copied to a IIgs disk and used, but it was never released as a product. It fell into the limbo of copyrighted products that were never released, and therefore can't be distributed legally except by the copyright holder. Chris Budewig (a.k.a. Kevin Flynn) has written a BASIC program titled **FONT.MUNGER** that can assist in converting a Mac font once you isolate the resource fork from the Mac file containing the font to a ProDOS disk using a utility such as John Wolf's **MacTrans**. You will have to supply some information to **FONT.MUNGER**, you can extract most of this information from the font resource using Chris' **RESLIST** utility. Chris' files include the basic instructions with the program. But if you want a font number from the **A2-Central** Font Clearinghouse, you will still need to contact Mark. -DJJ*

Affordable(?) laser printing

I was seriously considering purchasing a *LaserWriter SC* for my IIgs (just to play around). But I found out that it was not a *PostScript* device.

Would an older *LaserWriter* (used) at \$1895 be a better buy? Or a *LaserWriter Plus* (used) at \$2395? Or the *LaserWriter SC* at \$1980?

Joe Schmidt
West Milford, N.J.

Do you recommend the *LaserWriter NT* to print with *AppleWorks GS* and *Publish-It! 2.0*, or should one wait and see what develops in terms of higher-quality printing?

Horst Moehrke
San Diego, Calif.

We have been (very happily) using a **LaserWriter NT**, but the price will scare off most people. We have seen a \$695 cartridge advertised for the Hewlett-Packard **LaserJet** series II printers that supposedly adds "PostScript" (Aldus's page description language) compatibility using Phoenix Technologies' **PhoenixPage** interpreter and 35 **LaserWriter NT**-equivalent fonts; the product is called **PacificPage** and is sold by Pacific Data Products, 6404 Nancy Ridge Drive, San Diego, Calif. 92121, (619) 552-0880.

We can't recommend the **LaserWriter SC** for multiple reasons; the lack of a IIgs driver, lack of **PostScript**, and lack of AppleShare network support. The **LaserWriter SC** also depends on the host computer to do more of the work in laying out the page image; which requires more of the host computer's (RAM and processor time) resources. At about \$2800 retail for the **LaserWriter SC**, the **LaserJet/PacificPage** combination looks like a more attractive investment, if it works. A used **LaserWriter** or **LaserWriter Plus** is also more attractive.

There will no doubt be better (and hopefully cheaper) high-quality printers in the future, but buying decisions have to be made in the present. It breaks down to looking at the features and deciding if they justify (to you) the cost.

PostScript files and you

How do I save an AppleWorks GS file so that I can print it at the local printers shop that uses a Macintosh and a LaserWriter? Will my page

layout be thrashed? Will I lose all graphics images? Will the fonts be the same?

Alan Hansen
Des Moines, Iowa

The Apple IIgs **LaserWriter** driver will generate a text file of the **PostScript** program needed to produce your printout. You can then download this program to the **LaserWriter** later to generate your document.

First you need to install the **LaserWriter** driver on your boot disk. Then, from a desktop application (such as **AppleWorks GS**), pull down the "Apple" menu and select the Control Panel desk accessory. When the Control Panel window pops up, click on the **LaserWriter** icon. A window with "Select a **LaserWriter**", "AppleTalk Zones", and "User Name" will pop up. If you have a **LaserWriter** connected, select the printer and AppleTalk zone as usual. If you do not have a **LaserWriter** connected to your system, the "Select a **LaserWriter**" window will be empty; click inside it anyway to tell the Control Panel that you indeed want to use the **LaserWriter** driver. Enter a user name if the one shown is incorrect, and close the Control Panel window.

Next, select "Page Setup" from your program's "File" menu to make sure you have the page layout established correctly. Accept the settings when correct and, with your document loaded and ready to print, select "Print" from the "File" menu. The **LaserWriter** print dialog should appear. To create the **PostScript** file instead of sending the document to the printer, press and hold open-apple-F as you click on the "Print" button; continue holding open-apple-F until a dialog telling you that the driver is "Creating **PostScript** file..." appears. The file will be created in the **SYSTEM/DRIVERS/** folder, and will be given the name "POSTSCRIPT.GSxx", where "xx" is the number of the file being created ("POSTSCRIPT.GS01" for the first, and so on). These names aren't descriptive, so you'll want to keep track of the contents of each file and rename it to something recognizable (like "LetterToMom.PS") later.

There isn't a standalone **PostScript** file downloader for the IIgs presently, though Mark Collins hopes to have one completed before the new year. Since you plan to try and print the document from a Macintosh, you can use one of the Mac font/**PostScript** file downloaders such as the one Adobe supplies for downloading its (commercial) printer fonts. The place you go to print may have such a program available.

You'll also need a way to transfer the file from a ProDOS disk to the Mac. Apple File Exchange's "default" translation works; it's faster and simpler than the "text" translation since the file is known to be text the **LaserWriter** can deal with. Another option is modem or serial port transfer. If the IIgs, **LaserWriter**, and Mac are on AppleShare, you can just copy the text file from the IIgs to an AppleShare server volume that the Mac can also access.

The page layout problems are the same that you experience with IIgs programs in general; the converted Mac fonts most IIgs programs use print as tall, skinny letters unless you print using the "condensed" option. If your document also uses graphics, then the "condensed" mode will compress them to half their displayed height. Currently, there is no overall solution to this; the workaround is to "stretch" the graphic to twice its normal height with a

graphics utility program, then print everything condensed, or to not use condensed print and live with tall letters. (See the February issue for an explanation of this.)

Graphics imported (or drawn) as bitmapped images (images consisting of a series of on/off dots) will look like they are made of visible dots, somewhat as they appear on a dot matrix printer (admittedly, a very nice dot-matrix printer); you'll notice things like a "jaggedness" to diagonal lines. Graphics produced as "objects" (images described mathematically rather than as a pattern of dots) will print without the jaggedness. As an example, the drawing tools of the "Paint" module of **AppleWorks GS** produce bit-mapped images, while the drawing tools within the "Page Layout" module produce objects. This correlates to the Macintosh; "paint" programs (such as **MacPaint**) often are distinct from "drawing" programs (like **MacCAD**, nee' **MacDraw**) in this regard, though programs such as **SuperPaint** can produce either type of image. The capabilities of IIgs programs may change as they are revised.

The management of fonts is a little complicated; the rule of thumb is that if you want the best quality (with perhaps a little loss of "what you see is what you get") you should select fonts that exist in the **LaserWriter**. The original **LaserWriter** contains the Courier, Helvetica®, Symbol, and Times® font families as built-in fonts; the **LaserWriter Plus**, **NT**, and **NTX** also contain Helvetica Narrow, ITC Avante Garde®, ITC Bookman®, ITC Zapf Chancery®, ITC Zapf Dingbats®, New Century Schoolbook, and Palatino®. (fonts can be added to the **NTX** via an expansion board.) Additional fonts must be downloaded; Mark Collins' **PostScript** downloader for the IIgs will also be designed to handle fonts.

If the "laser" font (internal or downloaded) does not exist, a bitmapped generation of the font will be used. Checking the "Smoothing?" item in the "Page Layout" dialog box will cause the system to attempt to "smooth" the bitmapped image for the output. You can also elect to use the "Font Substitution" item to force the **LaserWriter** to use the internal fonts Times, Courier, and Helvetica in place of the fonts New York, Monaco, and Geneva respectively if the latter fonts are not present in the printer.—DJD

It's serious, folks

Thanks for your continuing terrific work. I have a IIgs, which I use extensively in my law practice (mainly business litigation). I wholeheartedly agree with your view that the Apple II can be a sophisticated business tool. I am typically involved in complex litigation, which requires amassing, sorting, and analyzing lots of documents and data, and then doing detective work (e. g. searching for intentional misrepresentations in financial statements, insurance company claim payment records, and the like). I simply could not do this sort of thing as easily and carefully without the IIgs. There is no good reason why Apple does not tell people that the II can do this sort of thing, and do it well; I think better than the Mac or the IBM PC's.

Gary Young
Madison, Wisc.

A2-Central™

© Copyright 1989 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	Steve Kelly	Jay Jennings
Tom Vanderpool	Jean Weishaar	

A2-Central—titled **Open-Apple** through January, 1989—has been published monthly since January 1986. 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 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 unutilized 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.