

A journal and exchange of Apple II discoveries

Some substance on style

by Dennis Doms

Putting style over substance (also known as form-over-function) creates a sore point with many people. Those who emphasize style are often condescending toward those they consider "artistically illiterate." Those who emphasize substance are outraged by the concept that something becomes more meaningful just because it's "prettier."

A2-Central has been through this for years. In times of multicolor and graphically-intensive (sometimes overloaded) publications, we have stuck to the concept that it's the message our readers are primarily interested in. Putting substance first embodies communication, which in our eyes is good; putting style first represents advertising, which we hold in lower regard. Advertising may convey useful information (we hope ours does) but communication of a specifically slanted message (that is, "buy this from us"), not empirical truth, is usually its primary purpose.

Subtly, we do incorporate style among the substance where we hope it aids delivery of the message. We use the best typesetting methods we can find, which allows us to quickly and economically bring our publications to you each month. Although we've touched on the mechanics of desktop publishing many times, this month we're going to pull together in one place the more basic style tricks we use that are applicable to the Apple II and especially to IIGs desktop applications that use more than the basic ASCII character set.

Step 0: learn to write well (or at least competently). This is the tip that many desktop publishing books skip, possibly on the assumption that anyone who has invested money in desktop publishing equipment is already an accomplished writer. Unfortunately, this is about as safe as assuming that anyone who buys a high-performance car can drive like Mario Andretti. The highways are littered with such people (and, unfortunately, their innocent victims).

For any young people out there, let's say it now: the best time to learn is in school. Pay attention in English classes (learn and relearn the basics until they jump out of you unbidden when needed) and in literature classes (notice why a piece of writing is special enough to have survived though the years). If you hit a lull in your required courses, fill your schedule with an extra English course.

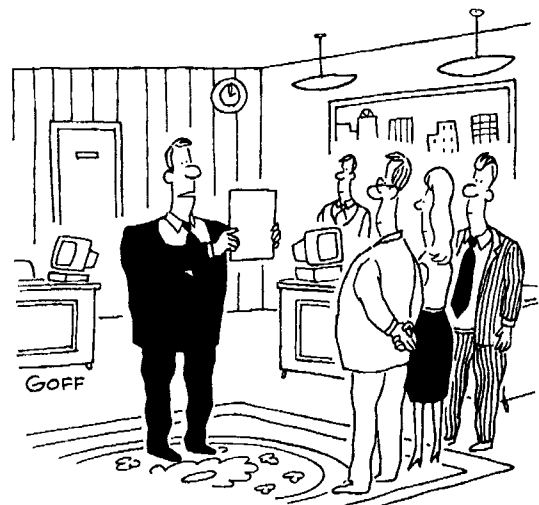
I scorned English through most of my years in school; my major interests were science, math, and music. Then I had an 11th grade English teacher who managed to get it through my thick skull that someone who wanted to be a scientist was going to be pretty frustrated unless he could communicate all those anticipated brilliant ideas effectively. I took an extra year of English and a course every semester my first two years of college. During at least one job interview, I was specifically told that the greatest factor in my favor was my demonstrated ability to write detailed and comprehensible reports. Thank you, Ms. Searcy.

If you now feel like you squandered some of that opportunity in school and it's too late, it isn't. Most communities have continu-

ing education classes; find them and attend them. Or if you're the self-help type, find some good references and work through them. This needn't be expensive; they have these wonderful information repositories called "libraries" that you are probably paying taxes to support. You should reap their benefits.

It's always good to practice; if you feel you need to comment about something, form the ideas in your mind and get them onto paper (or into your word processor). My personal belief is that the telephone is an inept way to communicate important ideas other than those that are obviously time-critical or require timely interaction with another person. Writing gives you the opportunity to review and revise your message. We should use the telephone when the message is simple and we need to get it to the right ear quickly, or for interaction, but not when it is an obviously inappropriate medium. It's amazing how many people will try to use a phone conversation to conduct extremely complicated and visually-oriented tasks such as debugging software; it simply is excruciatingly difficult to do without a visual reference. This is especially true now that faxes and electronic communications make delivery of written materials nearly instantaneous. Of course, if the missive isn't carefully and lucidly constructed, the speed at which it's delivered may be inconsequential to the rate at which its message can be extracted.

Once you have developed the tools, try to always apply them; make good writing habitual. Also keep ancillary references within an arm's reach any time you sit down to write; a grammar text (*The Elements of Style* by William Strunk and E. B. White is an excellent concise reference), dictionary, thesaurus, and any other style guides you may need regularly. For example, our publisher Tom Weishaar



"WHILE OUR SOFTWARE IS BEING DEBUGGED,
WE'LL ALL BE USING THIS SUBSTANCE
CALLED PAPER."

has deemed *The Associated Press Stylebook* to be our standard for most decisions here at Resource Central. If you make an error, make it because you forgot or were lazy (my two nominal excuses), not because you just didn't know any better.

Use computer-based tools if available, but don't depend blindly on them. Specifically, most modern word processors support on-demand dictionary and thesaurus functions that are usually acceptable, but I've yet to see a grammar checker that does as well as an attentive human at the controls.

I also think it's important to keep reading. I've heard comments that many adults have not read a complete book since graduating from high school; this is frightening. Encouraging your community to sponsor activities that enhance literacy will bring benefits to all of us; a literate society that is able to communicate ideas freely simply functions better.

Step 1: Concentrate on the content of your material first. If the idea is truly important it needs to be fully developed first. After you know what you need to say, deciding how to best present it can be approached in an informed manner. (A 40-foot billboard with bikini-clad women seems to be a popular approach, but in a more perfect world, maybe the message should garner some of the effort.)

This becomes more important as our alternatives for communication multiply. The very computers that have allowed me to write for a living (I can't type on a typewriter to save myself) also give us many new options for presenting our ideas in combinations of words, graphics, and sounds as well as for disseminating them. Hypermedia, desktop publishing (where you can compose a document and have thousands of copies ready to send out the next day), and information networks (where you can often send your ideas around the world in a matter of minutes) have given us the ability to say things in a wide variety of ways and have them get to others **quickly**.

Those of us who ride the electronic ether of networks like GEnie know all too well that occasionally you spot the most grievous errors in your document immediately after issuing the "send" command. The responsibility is more apparent now than ever to think before you let your thoughts escape your grasp; although you can try to alter or enhance interpretation of the message later, often the first perception is the one people inexorably hold. Don't let it be the wrong one.

If the content of your material is mainly the written word, then we have some tips for making it look nice in print. We'll leave hypermedia and other means to others for now.

Step 2: enter the text in the correct manner for a computer. Initially you have to make general determinations about how the text as a whole is to look, just as if you were typing it. On the other hand, you have to "fix" some of the typewriter habits before you start entering text into your word processor to make things easier later on. Back on the subject of necessary references, Robin Williams (no, she's not the actor and comedian) has two books available through Peachpit Press (1085 Keith Avenue, Berkley, Calif. 94708, 800-283-9444, 415-527-8555, fax 415-524-9775), entitled *The Mac Is Not A Typewriter* and *The PC Is Not A Typewriter*, that address most of these issues in a concise, orderly manner. The Mac version is mostly appropriate for Apple IIgs desktop applications, and some of the tips carry through to Apple II text documents as well.

Most of us are aware that typing on a personal computer is much different than a typewriter. Setting up margins, tabulations, and so on is completely changed, but even simple typing can trap you. One of the major differences is the RETURN; on a typewriter you hit RETURN at the end of each line. If you do this on a computer, you'll force ugly things to happen if you try to reformat your document later. So on the computer, you only press RETURN when you intend to end a paragraph. (Like this one.)

An example? Here's a "paragraph" broken with a carriage return at the end of each line, formatted with the same margins but in two different type sizes:

The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog.

The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog.

Our wonderful word processor nicely "rewrapped" whole words so that they stayed intact and didn't overrun the margins. The trouble is that each "Return" entered at the end of a line looks like an "end of paragraph" to the word processor, so our four original lines now look like four little paragraphs. If we use a carriage return only at the end of the text, it will rewrap without incident as the font is enlarged:

The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog.

Another problem is the use of two spaces after a period, which people always seem to argue about. Williams clarifies that this practice was started to emphasize the end of a sentence in monospaced fonts (like Courier) where all the characters are the same width. Look at the following paragraphs in Courier (with two spaces after punctuation) and Benguiat (with one space):

The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog.

The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog.

Proportional fonts like Benguiat assign different sizes to letters corresponding to their relative shape; therefore in a proportional font "i" will always be narrower than "m". The purpose of using two spaces at the end of a sentence on a typewriter (which uses a monospaced font in which all characters have the same width) is to visually distinguish the break between sentences. Although not as pronounced, proportional fonts already allow for this in the way they space characters.

Williams has several other points. One of the most important, which we often see broken, is to always use *tabs* instead of *spaces* to format text. With many computer applications, tabs are usually defined as distances instead of character counts. If you use spaces to separate text, the relationship may change if you later change the style or font (which changes the size of the characters). What looks fine in Courier:

A pseudo table in Courier with tabs.
A pseudo table in Courier with spaces.

a simple font change (to Benguiat, below) can destroy in a second:

A pseudo table in Benguiat with tabs.
A pseudo table in Benguiat with spaces.

Another point we brushed by in answer to a letter entitled "Apple-Works international" in the May 1990 issue of **A2-Central** (pages 6.28-6.31) concerns extended character sets. In particular, the extended character ASCII set was shown in the inset box on page 6.29. The text mentions how to generate several of the international

characters and also refers back to the letter, "More keyboard options" (February 1990, p. 6.7) for information on the option-prefix key sequences. Not only do these characters exist, they should be used where appropriate. The IIs fortunately uses the same keyboard commands as the Mac, so William's book turns out to be useful for this information; just don't count on having all the characters available in all font sets, as there are exceptions. And, if you're using Westcode's *Pointless*, you'll need to remember to enable the use of "extended" characters (those with ASCII values above 127 decimal) to generate these characters in your type. Among the most common ones you'll want to use (and how to type them from the keyboard):

Option-[and Option-]	"Curly quotes"
Option-) and Option-(	'Curled single quotes'
Option - -	- the hyphen
Option - _	— the long dash
Option-8	• the round bullet
Option-2	™ the trademark symbol
Option-R	® the registered trademark symbol
Option-G	© the copyright symbol "©"
Option-;	...the ellipsis

There are lots more (Greek symbols, foreign symbols, and a few more special characters). The ones we use the most are curled single and double quotes, hyphens, long dashes, and ellipses; in fact, many Mac programs support generation of these automatically, substituting the "smart" versions when you type the "dumb" ones (and we'd like to see IIs programs that did this, too).

After you're done typing, *proofread*. Check the grammar, the punctuation, the spelling...everything. Actually, you should probably proofread the document after any editing revision. Every time through you may find another error; this reduces the number you'll find when you see your work in print (there are always more).

Step 3: Work on the overall formatting. Once you've got the text typed in, you'll want to start working on the visual appearance; this is the "desktop publishing" phase. I prefer to work from the narrowest level (what specific words need stylistic emphasis) through the intermediate level (emphasizing groups of words or lines such as headings, program listings, and so on), then finally adjusting the layout on the page (where to put text boxes, inserted pictures, and other items that "break" the continuity of the text).

This is more of an "artistic" endeavor, which means hard and fast rules aren't as easily summarized. Williams has some tips for the more fundamental points of style; some match what we do, some don't. Whatever rules you come up with, be consistent; even if your rules vary from what someone else deems "acceptable," your readers will adapt if your style is consistent.

At the lower levels, *italics* and **boldface** usually tend to be the emphasis methods of choice. Underlining, which is often one of the few options available on the classic typewriter, is less used on the computer (Williams points out that underlining breaks up the "form" of the word and may interfere with readability, making it less desirable). Williams also points out alternative methods such as varying the **size of the font** or using SMALL CAPS, but we've tended to avoid these. (The tips and the reasons for offering them are good to read about, though.)

What you can do with the formatting tends to vary widely with what program you are using. You're going to be more constrained in a word processor such as AppleWorks than in a desktop publishing package such as *Publish-It!* or *GraphicWriter III*. The best recommendation I can think of is to pick up a book on desktop publishing design. Regardless of what program you use, you'll be able to adapt the book's advice to the program you do use. For example, you can't easily do wildly different font sizes in AppleWorks, but you can use a bold double-wide font for headlines versus a more normal font for the text in the body of your document. Also, general advice, such as interspersing white space to create pages without wall-to-wall text like we

normally have here, can be applicable no matter what software you use.

Even without desktop publishing, the concepts are useful. AppleWorks doesn't normally do multi-column printing or insert graphics (though there are add-on utilities that help with these), but if nothing else, you can produce your text with AppleWorks and then "cut and paste" it into a new form. I did this for a newsletter years ago; printing out narrow columns of text in varying font sizes, cutting them into the necessary lengths, and then pasting them up into two- or three-column pages with interspersed graphics, headlines, and other items. To make the document fancier, visit your local art supplies store; they have artwork, borders, and the sharp knives, paste-up board, and tacky adhesives you'll need. (Being unable to draw a straight line, one thing I found very handy were rolls of very narrow black "tape" which could be layered onto the page to create "rules" that separated articles, defined columns, and surrounded graphics.)

Personally, once past the necessary basis of good writing, I think Williams' book is a great place to start. Not only does it begin with the text (which is supposed to be the important part) but it doesn't go too far afield into the overall design of something like a newsletter. Learn its lessons well, then find a book on document design to complete the cycle.

Tips on Documenting Software

by Phil Shapiro

The difference between good software and excellent software often has nothing to do with what's shipped on the magnetic disks. In many cases the software succeeds or fails based on the accompanying documentation. Here are some tips to consider when writing documentation.

The time to start thinking about documentation is before you've written your first line of code. By thinking about documentation concerns during the design phase of development, you may be able to save yourself time and effort in writing the docs. Keep in mind the ideal of all software: so simple and intuitive that you have no need to write instructions on how to use it.

Documentation need not be lengthy and detailed. There's a common misconception among independent software developers that all commercial software requires detailed and lengthy documentation. Not so. The documentation for one of the most successful educational games for young children, *Facemaker Golden Edition*, consisted of nothing more than two folded pieces of paper. Spinnaker sold more than 150,000 copies of this software.

Moral? The software was so well designed that it hardly needed any documentation.

Consider who will be using your software. Consider their average level of computer experience. Write documentation to fit this mythical "average person's" needs.

The best way to learn how to write documentation is to study the successes and failures of existing documentation. Spend time consciously studying the organization and layout of documentation booklets from major software publishers. There is no reason why you can't borrow their major subject headings in your own documentation (should you determine the headings would apply to your own software.) Likewise, there's no reason why you can't lift short explanatory words and phrases from documentation you study. Jot down phrases that you think might have relevance in describing your own software.

Critically examine all documentation you read. You can learn from other people's mistakes. Ask yourself how you would do things differently if you had to document that particular piece of software.

Logic would suggest that the main programmer for a software project would be the best person to write the documentation. Not always. The main programmer for a software project has

such an intimate, close-up view of the program that he or she might not be able to see the pitfalls novice users will encounter. Too, the main programmer for a project should theoretically expend all his or her energy in creating the best possible software product. Writing up award-winning documentation as an encore is expecting too much of one person.

(Notable exception: Paul Lutus, after creating the *Apple Writer* word processor, proceeded to write his own excellent documentation. Other notable exceptions are sufficiently numerous as to cast a shadow of doubt on the previous paragraph's recommendations.)

Imagining an audience for your documentation. Writing documentation requires that you imagine an audience for your documentation. A helpful technique is to imagine that you're writing an explanation for a good friend who is interested in using the software you developed. Imagine that your friend is relatively new to computers, but does not need to have his or her hand held every step of the way. If you have a good friend who meets these "imaginary friend" requirements, so much the better. Send him or her a rough draft of the docs, and ask for rigorous feedback. (Soliciting "polite" feedback will serve you no use when you go back to edit the rough draft. If your friend can't tell you every minute flaw in the docs, find a more honestly critical friend.)

You've got to be very cautious about using humor in documentation. Since humor is often culture specific, the chances for humor backfiring are substantial. Consider this quote from Apple's DOS 3.2 Manual (explaining an error message): "What happened was this, your Apple II went on a fruitless, unending search for information on a blank diskette (on a clear disk you can see forever...)" Cute? Perhaps. But imagine the frustration of a non-English speaker trying to make sense of this sentence and its obscure allusion.

Then again, Beagle Bros gained their large, spirited following by including humorous captioned wood-cuttings in their documentation. You may have noticed that as the company's products started reaching a more mainstream market, the humor rapidly tapered off. Moral? Humor worked well with the hobbyist crowd, but served as a liability once the products started penetrating the school and "professional" home market.

For further information on this subject you may want to consult *Writing Better Computer User Documentation: From Paper to Hypertext (Version 2.0)*, by R. John Brockman, published by John Wiley and Sons, New York, 1990. It is a very well-written book that is considered a classic in the field. This book considers everything from online documentation to typographical choices in hard copy docs. The bibliography at the end of the book includes an exhaustive 33 page listing of books and magazine articles on computer documentation. The author is an associate professor of computer documentation at the University of Delaware. He has lectured widely on the topic of computer documentation at colleges and universities. In 1986 he was the winner of the RIGO award for major contributions to the knowledge and understanding of software documentation.

(The author is the founder of *Balloons Software*, an Apple II educational software company. *Balloons* currently publishes *Number Squares*, a disk of fun logic puzzles, and *Big Text Machine*, a creativity toolkit. You can reach Shapiro on GEnie at: P.Shapiro1, America Online: pshapiro)

The Lost Tribe: a review

by Ellen Rosenberg

The Lost Tribe is a found treasure in the world of Apple IIgs software. Created by Lawrence Productions, previously acclaimed for the McGee preschool series, I found it to be an exercise in fun, learning and most importantly, the building of leadership and decision-making skills. What better qualities to prepare children for adulthood!

The Lost Tribe is set in prehistoric times. You play the role of the person appointed to guide your tribe to an ancient homeland after an unfortunate volcanic eruption on nearby Belchfire Mountain demolishes your village and kills the tribal leader along with most of the experienced hunters of the community. As the name of the volcano might indicate, *The Lost Tribe* is not only educational but cleverly written with lots of humor interspersed throughout the factual information.

As the software boots up, you are treated to the sounds of primitive drums, beating softly in the distance. The music intensifies and your screen gives way to the image of the remaining members of the tribe wandering off in search of their forgotten homeland. You choose one of the six difficulty levels and you're off to meet your fate. The future of your kinspeople lies on your shoulders and leadership skills.

During the course of the adventure, you must make many choices that will affect the morale and well-being of the tribe.

Typical types of choices involve how quickly to travel, how much to eat, whether to hunt for food or make carvings for luck. Random events occur that force you to make intelligent decisions for your people or settle (or not settle) petty squabbles between tribe members. If you make your decisions wisely, you'll be rewarded with a happy tribe and a new found homeland. If not, you're banished from your tribe in shame.

Reading the 100-page online illustrated encyclopedia will greatly improve your chances for winning the game. Winning aside, it's so informative and such interesting reading, it's a part of the game that shouldn't be overlooked. Topics covered include personality profiles of the tribespeople, land formations, prehistoric animals, leadership qualities, tribal societies, tribal legacies, art & artifacts, tools and utensils, prehistoric trivia, and ancient mysteries. Each of the ten topics include 10 sub-topics. The ancient mysteries section touched upon such issues as creation versus evolution, the disappearance of the dinosaurs, pole shifts, and vortex energy.

I found the Leadership Qualities section particularly valuable not only for help in winning the game but for the teaching of life skills as well. The ten cited qualities should be made into a subliminal tape to be placed under your child's pillow at night and she will surely grow up to be senior class president.

The creative teacher and/or parent can find a wealth of information to spark extended activities for classroom or family projects in the pages of this online reference. For instance, the Alphabet section contained in the Art and Artifacts section, showed comparisons among the English, Mayan and Egyptian alphabets and encouraged comparisons to be explored between other alphabets around the world. What a great idea for an individual, or a small group project. The section on crystal skulls was so fascinating (and a subject with which I was totally unfamiliar) that I plan on researching them more thoroughly.

The Lost Tribe comes on two disks and is not copy-protected.

It's hard-drive installable and requires a minimum of 1 meg RAM but 2 megs is recommended for ease of use. In addition to the well-planned manual; albeit aimed at the MS-DOS and Mac market, there is an insert and a Read.Me file on the disk that are specific to the Apple IIgs version. They do a good job of explaining how to run the program on various systems; including those without a hard drive. There are over 80 scanned photographs (in the IIgs' 256 mode) of the clan in action, animated cartoonish scenes depicting them at various tasks throughout the day and digitized music to add to the experience. *The Lost Tribe* retails for \$49.95 and is available from a number of mail order houses including Big Red Computer Club. It's recommended for children aged 8 and above. This adult obviously enjoyed playing it too! *The Lost Tribe* is a unique edu-tainment addition to the Apple IIgs line that shouldn't be missed. For further information contact Lawrence Productions, Inc., 1800 South 35th, Galesburg, Mich. 49053, 800-421-4157, 616-665-7075, fax 616-665-7060.

Miscellanea

There's nothing like San Francisco in the springtime (well, maybe Paris, but I wouldn't know). Not that we really got to experience San Francisco in the short time we were there but the Apple Expo West on April 22 and 23 sure was a treat. It was a great place to network, to see new products and get some good bargains.

There were lots of Apple II companies and organizations manning booths. I'll attempt to list them for your information. Please forgive me if I've missed anyone: A+ Publishing, Alltech Electronics, BAAUG (Bay Area Apple II User's Group), Big Red Computer Club, DigiSoft Innovations with Lunar Productions, Econ Technologies, Educational Resources, Gravenstein Apple II's User's Group, Learning Services, NAUG, Parson's Engineering, Procyon, Quality Computers, Resource Central, RezTech, Roger Wagner Publishing, Sequential Systems, Seven Hills Software, Simplexity Software, Softdisk Publishing, The ARC, Ventura Educational Systems and Vitesse.

One of the more interesting things to note was the fact that there seemed to be no bad vibes between the devout Apple II and Mac factions at this show. Everyone appeared to get along (well, almost everyone) just fine, no bad mouthing other platforms, no dirty tricks. Now if the world at large could do the same....

Ventura Publications, whose booth was across the aisle from ours, is an educational-oriented company that was previously unknown to me. They carry many products for Apple II users that looked really interesting. Naturally, they were educational in nature; great for parents to reinforce skills at home. One product that would be of interest to the general public was a project-based HyperCard tutorial. One package was aimed at children (*HyperCard Projects for Kids*) and one at teachers (*HyperCard Projects for Teachers*). The books contain over 100 pages and include a disk of templates and operational stacks. The disk and book are aimed at the Macintosh market, but converting the disk to IIGs format should be easy with HyperMover. For further information and a catalog contact Ventura Publications (910 Ramona Avenue, Suite E, Grover Beach, Calif. 93433 or call 800-336-1022.)

The sun is definitely dawning on DigiSoft Innovations' Twilight II. It's one of the new products I found *finally* released at the Apple Expo West. This new desk accessory functions as a screen saver and what a screen saver it is! It works with just about any program you can use with your Apple IIGs, including text-based programs

like AppleWorks. It comes complete with a detailed manual and so many modules you'll probably lose an entire afternoon deciding which one you want to use. Two of my favorites are Fireworks and Universe. Fireworks is, you guessed it a dynamite fireworks display, complete with sound effects that would rival just about any July 4th extravaganza anywhere. Universe is a black and white intergalactic space ride through the galaxy. Twilight II is affordable; it retails for \$39.95 and is carried by a variety of resellers including Resource Central. Upgrades are available directly from DSI (Box 380, Trumbull, Conn. 06611).

We've gotten a good number of letters from people wondering how to connect their ImageWriter II's to their PC clones. While we don't have any direct experience with this, I found a message on GEnie, reposted from the Internet by Lunatic E'Sex, that might be of interest. I've edited out the Internet headers and reposting of the original questions, leaving just the answers. Here goes:

"I hooked up my ImageWriter I to my COM1 serial port and used the C-Itoh 8510 driver in Windows 3.1. The cable I am using is a straight-thru cable connected to a Null Modem Adapter I got at Radio Shack (catalog #26-1496a) for \$4.95. It seems to work fine with both DOS and Windows. I used the following command in DOS:

```
MODE COM1:9600,n,8,1,p
```

and set up the port the same way in the Windows Ports setup. The Null Modem connections are as follows:

1	to	1
2	to	3
3	to	2
4	to	5
5	to	4
6+8	to	20
20	to	6+8
7	to	7

I printed from several applications and all seems OK."

"You should use a hardware-handshake RS232 cable available at many computer outlets. If you only have a regular serial cable, make sure you use XON/XOFF handshaking. You need to change a IW2 DIP switch for this; check your manual. Don't forget to set your serial cards to XON/XOFF.

Normal text is no problem. If you want to make use of 'special features' on the DOS end, you need a suitable driver. Wordperfect 5.1 has a driver for the IW2, I'm not sure about other programs."



Ask (or tell) Uncle DOS

Confessions of a hyper-dweeb

Four years ago if someone had told me that I would be editing a computer disk publication and making and selling my own software, I would have laughed. Today I'm doing just that and having the time of my life to boot. The surprising part is I never meant for it to happen. It just did. Four years ago I was happy enough

collecting records (remember those things?) and video taping just about anything that showed up on television. Those were my hobbies and I was more than happy to maintain a status quo, but all of this came to a screeching halt on Christmas Eve in 1989.

The computer entered our home the way it does for many middle-class American families. Here's the pitch... "You're children will become higher achievers with a computer at home. They can run all of their favorite programs from school. You can keep track of your family finances and build a database containing your Christmas card list or all of your favorite recipes." Friends proclaimed I could catalog my record collection and have all of that valuable information at my fingertips. Wow! Where do I sign?

As it turned out we qualified for a generous discount from Apple because my wife, Betsey, is an educator. So late in November of 1989 I brought home a brand new Apple IIGs system with an ImageWriter II, carted it up a very shaky

set of attic stairs (*Let's surprise the kids with it on Christmas Day!*) where it sat until "...not a creature was stirring; not even a mouse." I have to confess I did make one raid into the attic before Christmas Eve. Late one night I opened every carton and collected all of the documentation (weighing in at about 15 lbs.) so I could have a jump on things when it was time to set up. When the appointed hour arrived I brought each carton down, carefully unpacked its contents, put it all together (*Boy! Am I glad I read the manuals in advance.*) and sat down to my first real experience behind a computer. It took me until 3:00am to make an acceptable Christmas tree and surprise greeting using *Paint-Works*.

For the next year and a half I bought and used the usual variety of software, played dozens of games, peeked and poked around with AppleSoft BASIC, spent hundreds of dollars and generally lost contact with the entire family. By that time we had no database of Christmas card addresses, no collection of

favorite recipes and our family finances still consisted of papers, bills and checkbooks stuffed into a desk drawer. What's more, if the kids brought programs home from school they had about a snowball's chance in hell of getting computer time because dad was "working."

Nothing about 1991 hinted that things were about to change until my birthday rolled around in February. A friend had sent me a used 1200 bps Hayes modem and the family presented me with *HyperStudio 2.1*. I was online that same night. *HyperStudio* sat on a shelf for two more months. When the fascination of telecommunication began to wear off, I needed something else to "get high on" (a software fix!). That's when *HyperStudio* was rescued from the closet.

My earliest attempts at stack design were, of course, crude and tentative. I played around with paint and animation (by then I had become a real power user with *PaintWorks*) and after a few false starts I completed a stack that I named *Stupid Button Tricks*. This was a little collection of cel animations that included a pair of windshield wiper blades, a disk ejecting from a drive and a few others. Feeling quite satisfied with this effort I uploaded it to the Apple II RoundTable on GEnie and the Apple II Forum on America Online. No big deal. I made another, called it *More Stupid Button Tricks*, and uploaded it.

Around the same time Dean Esmay, GEnie's Apple II software librarian, became the new editor of *Stack-Central*, the *HyperStudio*-based disk magazine from Resource Central. With the deadline for issue #9 rapidly approaching Dean plucked my two stacks from the library and gave me a call.

Dean: Hey! Can I put your *Stupid Button Tricks* on *Stack-Central*?

Bill: Are you kidding?

Dean: No. I'll even pay you for them.

Bill: Who is this *really*? (banging the phone on the wall)

And so began my affiliation with Dean and *Studio City* (nee *Stack-Central*). Today I am submissions editor for *Studio City* and my stacks have been featured on every issue since #9. I'm having the time of my life and I'm getting paid for it. But that's not the best of it.

Since early on, one of my goals has been to learn enough programming to allow me to make software for physically disabled computer users. With 23 years of experience in special education, it occurred to me that I was in a unique position to influence software design for these special needs. Before I ever sat in front of a computer I was tinkering with gadgets that could help disabled kids interact more effectively with their environment. Electromechanical switches, adapted toys and appliances, communication devices, all of these are valuable tools in the area of assistive technology. With my experience and insight surely I could make a contribution to the growing body of special needs software. There was just one thing stopping me. I wasn't a programmer. So I turned to what I knew best, *HyperStudio*.

With the help of a very special Xcmd (now an NBA) from Roger Wagner Publishing, Inc. I put

together a few primitive stacks that could be accessed using any electromechanical switch and an inexpensive Apple II switch interface. Briefly, the switch interface plugs into the game port and the user's switch acts as one of the fire buttons on a joystick, or as the APPLE or OPTION keys. I offered these stacks to a number of teachers who used the Apple IIos with their disabled students. They used the stacks, offered suggestions for improving them and helped the stacks evolve into valuable teaching tools. The advent of *HyperStudio 3.0* with SimpleScript gave my stackware a significant boost. Now the stacks could record, track and analyze the switch user's progress and provide hard data for the teacher.

Last year I began sharing my ideas on switch accessible stack design with parents, teachers and friends across the country. The notion of teacher-authored software is beginning to catch on thanks to *HyperStudio* and other authoring systems. This year I take my show on the road to local user groups, regional special education seminars and national and international conferences on assistive technology and adaptive computer use. In between travels I hope to complete my first print/disk tutorial entitled *Switched-On HyperStudio: Designing and Creating Switch Accessible Software*.

Over the course of nearly four years I have managed to turn an amusing pastime into a new career. I have made numerous friends along the way and I feel better about what I do than ever before. I'm still not a programmer (although my experience with scripting is beginning to introduce me to some rudimentary programming principles), I still don't have a database of Christmas card addresses, recipes or record albums, our family finances are still stuffed into the desk drawer and my kids are still lucky to get a half hour of computer time before I need to get back to "work". But I'm having a ball!

Bill Lynn
Northfield, Conn.

If you're interested in learning more about Bill's adaptive stackware or his monstrous Beatles stack, for that matter, you may contact him at Simtech Productions, 587 Northfield Road, Northfield, Conn. 06778.

Inclined to telecompute

In the early '80s I started my computer adventure using a Bell & Howell Black Apple II-Plus using a lackluster set of applications made by PFS to manage my business. About '83 I watched an Apple dealer demonstrated *QuickFile* on an Apple IIe (impressive) and *Three Easy Pieces* on an Apple III (not impressive). I sold the "Black Apple," bought the Apple IIe system, one drive, 128K and *QuickFile*. For \$3,000 I had started on a venture that was to change my life — what a deal. With no experience making or using data bases, I was able to easily set up my business inventory using *QuickFile* and I ran an inventory system which I and my peers felt was state of the art. Also during that time I was looking for a word processor. Magic Window was the rage in our user group.

Was I impressed — NOT!

I read a review of *Apple Writer* by Don Lancaster, was convinced, and purchased the DOS 3.3 version of the program. About a year later I upgraded when the ProDOS version was released. *Apple Writer* always gave me the feeling I was working with a fine tool. It's a wonderful and powerful word processor. I was able to use its Word Processing Language (WPL) to do some satisfying work, including making *QuickFile* and *Apple Writer* work together to write form letters and place a customer's name/address info on queue anywhere in the letter, really impressive and a great combination!

Then AppleWorks was announced and *QuickFile* was the data base module. I thought WOW(!) it will be great to have *QuickFile* and *Apple Writer* in the same software — WRONG! AppleWorks not only did not have a word processor as good as *Apple Writer*, (and still doesn't IMHO) but the (*QuickFile*) data base did not work together with the word processor like *QuickFile* and *Apple Writer* did separately. I would be interested to know the politics involved for Apple to have made such an incredible mistake.

AppleWorks creates a proprietary file with extra overhead while clean ASCII text files are universal to just about all computers and software. Text files can be uploaded to and received from any online service and they are always just text files. AppleWorks files are NOT clean ASCII text files and they must be "printed" to disk to convert to a text file. That places a return character on the end of every line, which is good for telecom use, except files are always in need of being edited or updated, or maybe the line length was not correct for different telecom systems, and getting the text files into and out of AppleWorks was/is painful compared to what it could have been.

As an aside, I also worked closely with *WordPerfect* and even beta tested many of their Apple II versions. At that time *WordPerfect* documents were really difficult to convert to ASCII text files. Later they came up with a method of writing text files as "large macro" files, but by their own admission it was not a good fix, nor adequate for the average user.

Apple Writer saves files in clean ASCII text and files can easily be saved with or without return characters on every line. Just what was needed in a world where *ASCII Express* was the optimum telecom application (what an incredible statement by today's standards).

Apple Writer continued to be my favorite tool even after *AppleWorks* was released. In early '86 I asked Guy Kawasaki (when he was Apple's Software Evangelist) if *Apple Writer* could have some of its problems updated. Even in 1986 Guy was a totally a Mac person and had no interest and limited knowledge of Apple II software. His answer was a terse, "We don't fool with software that sells 2,000 units!" (The words are slightly paraphrased, but the meaning is the same.) I had become a sort of a power user of *Apple Writer* and I had a list of suggested fixes I felt would make *Apple Writer*

better. I communicated these ideas to Apple Computer, but they were not at all interested. *AppleWorks* subsequently dominated the market and *Apple Writer* was discontinued as a competitor to *AppleWorks*.

Later that same year (1986) I met Greg Schaefer (*ProTERM* author) and during that meeting he confidentially demonstrated *ProTERM* "barely out of the cradle." I knew I was seeing truly great software as it was being born, committed myself to be involved from the first *ProTERM* beta test and have been involved with *ProTERM* since. It was a given that *Apple Writer* was to never be what I thought it could be and as Greg developed the *ProTERM* Editor, I kept asking for features I liked in *Apple Writer* or would liked to have seen in *Apple Writer*. Over the years *ProTERM* has become even more powerful than *Apple Writer* as a robust word processor, especially if you are using the files for telecommunications. And *ProTERM* is **much** easier to use. Not bad deal considering you get a pretty good telecom application as part of the bargain <smile>.

Dot Commands are one of the many real powers of *Apple Writer* (a dot command consists of a period followed by two alpha letters). For example, a command such as .LM5.RM75 placed ahead of any text will cause the file to be printed with right and left margins to Left 5 and Right 75 respectively. The command .PM8 yields an indented Paragraph Margin of 8. Even hanging paragraphs can be set with a command such as .PM-5. .FF instructs the printer to Form Feed the sheet to start a new page. (BTW - *ProTERM* allows this and also Previews the formatted text on the screen as "What You See Is What You Get" (WYSIWYG).)

The bottom line is that *Apple Writer* is now free and still excellent! If you like *Apple Writer*, you'll **love** *ProTERM*. The similarities are remarkable but *ProTERM* is much easier to use, especially for the new computer user. Like the electric shaver advertisement by Victor Kiam, I liked *ProTERM* so well, I bought (into) the company.

If you would like to try *ProTERM 3.1*, a demonstration version is available by downloading it from the major online services or the InTrec BBS. We will also send it on disk if you prefer. Please provide an address and disk size preference.

Jerry Cline
InTrec Software, Inc.
(formerly InSync Software, Inc.)
3035 E Topaz Circle
Phoenix, Ariz. 85028-4423

The road back to the Apple II

I am one who find myself working with three different platforms almost daily (I know, I know, I should subscribe to *Fishhead's Children*). At the office, I am forced to work in the Messy-DOS environment and at home my trusty Apple IIgs with all the extras, occupies the space on the desk in my den. I had gotten to the point where I was going to have to start shopping for an MS-DOS clone so I could work on files at home in the evening. Much to my surprise, the

answer turned out to be a used Mac Plus with an external SuperDrive Clone (AE Plus drive).

After much research into file translation options, I found the most cost effective method was DataVis's software, *MacLink Plus* and the used Mac Plus. The Mac Plus acts as the middleman in the translation process. By running *MacLink Plus* and converting my Lotus 1-2-3 spreadsheet files to any Mac spreadsheet file and from there to an *AppleWorks 3.0* spreadsheet file. This method keeps all the formulas and data intact. The program will allow many MS-DOS files to be directly translated into *AppleWorks* files but not saved directly onto a ProDOS disk. It will first have to go to an HFS (Mac) formatted disk and then converted onto a ProDOS disk using *Apple File Exchange*, a program that comes in the box with every Macintosh. Then it can be read by *AppleWorks* as a spreadsheet file. Word Processor, database, and spreadsheet files are all easily converted to *AppleWorks* this way. I have found that 98 per cent of my files converted without a hitch. I also looked at *Cross Works* but found the conversion possibilities too limited. A few people at the office have Macs at home and I can easily converted MS-DOS files for them to use in their favorite Mac programs. The translation back to MS-DOS files is just the reverse of the first conversion process and again all the formatting and formulas are maintained. The *MacLink Plus* program has literally hundreds of file translators, however, the only Apple II translators are for *AppleWorks* (2.x, 3.x, & GS word processor files, 3.0 DB & SS files only).

Once in a while I'll need to transfer some raw data where I'm not concerned with the spreadsheet formulas. Thanks to System 6, this is a real snap. I take the MS-DOS data and convert it to a text file on the Mac using the *Apple File Exchange*, then I then take the HFS disk from the Mac with the text file and slip it into the 3.5 drive on the IIgs. The HFS FST recognizes the file and I save it to my hard disk. Now I've got an ASCII file on a ProDOS disk. This is loaded into an *AppleWorks* word processor file, copied to the clipboard, and then transferred to a new spreadsheet file. This method will work for importing data into *AppleWorks* GS also but I personally like the Classic *AppleWorks* spreadsheet better since I use *TimeOut Graph* to dress up my reports!

I really think this combination is the best all-around conversion system for the money. I bought the used Mac Plus with two megs and a 52 meg hard drive for under \$500. I bought the AE Plus drive for \$290 and *MacLink Plus* for \$109. As you can see, for under \$1000, I have a three-platform file exchange method that is fast, accurate and easy to use. I would really recommend it to anyone who refuses to quit using their Apple II for serious work and finds themselves in a similar situation to mine.

Steve Schmidt
Bethoud, Colo.

Tom has mentioned MacLink Plus a number of times in Fishhead's Children. His readers have praised it as well and have reported using it to recover files as well as convert

them. Keep in mind that it does file format conversions only. There are several ways to do disk format conversions, unfortunately, the current protocol isn't the same as the one used by Apple File Exchange. MacLink Plus expects AFE disk format conversions rather than network or HFS FST conversions, but DataVis is working on recognizing both.

Protection vs registration

I write to correct what I am sure was an inadvertent error in Jay Jennings' interesting and informative article in your March 1993 issue.

On page 9.11, Mr. Jennings asks whether one "need(s) to copyright a program before letting it out of your sight." He goes on to suggest that if you do not do so, "you won't be able to sue anybody who steals your program."

Mr. Jennings was confusing copyright protection with registration, since a computer program is copyrighted as soon as it is "fixed," which for practical purposes means written down on paper or saved to a disk, at the very least. Thus, if someone does "steal" your program, you can sue for copyright infringement, even if you had not registered it as of the date of infringement (though you would have to resist before filing the action). Even absent registration, your program would be protected by the Federal Copyright Act. In other words, you "must" register only before you sue.

The question Mr. Jennings probably meant to ask is whether you should register before letting the program out of your hands. If you do, and someone subsequently infringes, you would have a significant advantage over one who did not register before the infringement occurred; namely, in addition to the ordinary copyright infringement damage recovery, you would also be entitled to attorney fees or statutory damages in place of actual damages. Among other things, these prospects might make it easier to find a lawyer willing to take your case. If you had not registered before the infringement (even though you registered, as you have to, before suing), you would not be entitled to these remedies, although you would be able to bring a copyright infringement action for the infringer's profit or your own actual losses.

Steven M. Kipperman
San Francisco, Calif.

Sequential RAM card fix

Recently I became aware of on-going intermittent compatibility problems with our RAM-GS 4-megabyte memory expansion card and CV Tech's "side-car" style memory board. Several months ago, we replaced logic on-board with a higher speed variety to compensate for memory timing delays introduced by logic on the CV Tech product. Apparently, intermittent timing problems still exist in isolated instances. Effective immediately, in order to be as compatible as possible with other popular third party products, Sequential will make additional adjustments to compensate for these delays.

Please be aware that all Sequential Systems products are designed for maximum compati-

bility and that RAM-GS, in fact, has always satisfied all specifications for Apple IIgs memory upgrade as set forth by Apple Computer, Inc.

The complete satisfaction of our customers is of paramount importance to Sequential Systems and we will gladly replace any RAM-GS that may be incompatible with the CV-Tech board with the most current revision of RAM-GS. Fastest replacement can be made by calling 800-999-1717

Joe Yandrofski, President
Sequential Systems, Inc.
1200 Diamond Circle
Lafayette, Colo. 80026

Need Postscript help post haste

The ImageWriter emulator for the LaserWriter (v.1.5) has a nasty habit. When I initialize the printer for the German character set in BASIC or with AWP Printer control codes (interface or special codes) with the appropriate escape codes, the LaserWriter prints only the first page in German. As soon as there is a page feed, the US character set returns. Any idea how to change this? I had a look at the postscript file but did not understand it.

Gerald Steinbach
Gerlenhofen, Germany

A2-Central™

© Copyright 1993 by

Resource Central, Inc.

Most rights reserved. All programs published in **A2-Central** are public domain and may be copied and distributed without charge. Apple user groups and significant others may obtain permission to reprint articles from time to time by specific written request.

Publisher:

Tom Weishaar

Editor:

Ellen Rosenberg

with help from:

Denise Cameron
Dean Esmay
Jeff Neuer

Dennis Doms
Peter Haxton
Jean Weishaar

Sally Dwyer
Mary Johnson

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

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

A2-Central

P.O. Box 11250

Overland Park, Kansas 66207 U.S.A.

A2-Central is sold in an unprotected format for your convenience. You are encouraged to make back-up archival copies or easy-to-read enlarged copies for your own use without charge. You may also copy **A2-Central** for distribution to others. The distribution fee is 25 cents per page per copy distributed.

WARRANTY AND LIMITATION OF LIABILITY. We warrant that most of the information in **A2-Central** is useful and correct, although drivel and mistakes are included from time to time, usually unintentionally. Unsatisfied subscribers may cancel their subscription at any time and receive a full refund of their last subscription payment. The unfilled portion of any paid subscription will be refunded even to satisfied subscribers upon request. OUR LIABILITY FOR ERRORS AND OMISSIONS IS LIMITED TO THIS PUBLICATION'S PURCHASE PRICE. In no case shall our company or our contributors be liable for any incidental or consequential damages, nor for ANY damages in excess of the fees paid by a subscriber.

ISSN 0885-4017

Genie mail: A2-CENTRAL

Next to last time in TimeOut-Central

Directory: /TimeOut.Cent.15/

Filename	Blocks	Description
BUG.CITY	6	Bug report for AW users
CANADA.TAX	1	Granite Software's Ont Tax Analysis '92
CLEAR.NIGHT	1	TimeOut Cellmover
DAUG	1	
.PUSHING.CLIPB	8	Macro for TO PowerPack
.ORGANIZ.DEFMCRS22		Organize default macros
GENIE	1	Hacker Jargon
JEM.SOFTWARE	1	Use Add Files w/o grief
NELKEN.SERIES	1	Lessons 7 & 8 includes MadLib macro
PAYROLL	1	Calculation/posting of recurring payroll expns
SENSIBLE	1	Convert Sensible Speller dictionaries to QuickSpell/AW3.0 format
ULTRA4.2	1	Updates Ultra 4.1

Last time in TimeOut-Central

Directory: /TimeOut.Cent.16/

Filename	Blocks	Description
ROY.BARROWS	2	Series of macros by
LORNE.WALTON	1	Perpetual calendar
ANGELO.RUBINO	1	Quick Calendar macros
JEM.SOFTWARE	1	More great stuff from
NELKEN.SERIES	1	UM 4.xx lessons 9 & 10
ANGELO.RUBINO2	1	Help & hints for Super Font fans
DOUG.BROWER	1	Outlining in BASIC
CHRIS.SERREAU	1	AW Control Panel
TEXAS.II	1	News & top 10 offers
QUIZ	10	Quiz for TAPL prgmrs
ROD.YOUNG	1	
.CASE.CHANGER	15	Macro hilites & changes case of text
.T.O.D.	1	Test on disk--teachers, don't miss this one
DB.MACRO	1	Shrunk file of complex system of macros to link DB info
VERSION.CHANGER	3	
CLEAR.NIGHT	1	TO Clip Viewer: Sup ports viewing of clip board when DoubleData 2.0 is installed
BUG.CITY	7	Bug Report for AW users

Last time in A2-Central-on-disk

Directory: /A2.ON.DISK.9305/

Filename	Blocks	Description
Intro.May.93	20	Dean's blather
V9.N04.May.93	112	May issue
Letters.05.93	43	We get letters!
General.Stuff	1	
.Hayman.Confrnc	10	RTC w/ Magic File Cabinet author
.Raster.Blaster	54	Another Lost Classic-- Bill Budge's pinball
.Sys.Disk.4.0.1	209	8-bit System disk
.Notes.Sys.4.0.1	6	Apple's notes
IIgs.Stuff	1	
.Cute.Desk.Accs	84	Silly but fun ndas
.SkullCursr.Init	4	Change dull cursor to skull
.Manager.Compat	23	Compatibility list from Seven Hills
.ModZap.0.81	667	Great MOD player plus song
.ProBoot.5.2	144	Exceptional boot util- ity for GS/OS
.Spy.Hunter.GS	206	SpyHunter look-alike

Last time in Script-Central

Directory: /ScriptCentrl.12/

Filename	Blocks	Description
Script.Cntrl.12	17	Launch this first
WordScrambler	105	Find permutations of any word
FortunCookie2.6	595	More fortunes
firstLadies	337	Facts @ 1st ladies
ArtWerk2.6	124	School tools, lights flowers
HyperSonic2.6	117	More sounds
OverWeight2.6	64	Finished stack-deter mines body mass
xVolumes	20	Returns names of online volumes
OverWeight2.0	58	Your homework
xVolumes.SCR	1	Source code