GS+®

March
April
1994

Volume 5
Number 4

The *First* Apple IIGS® Magazine + Disk Publication!

MUTINY!

This Issue of GS+ Magazine Runs Red With the Blood of Our Oppressor!

Plus, These Fine Programs

LASERbeam v1.1
Playful
What Is This?

And These Fine Reviews

Addressed For Success
ORCA/Debugger vs. Splat!
ORCA/Modula-2

Writer's Block

By Steven W. Disbrow

Well, once again I'm trying to write this editorial a full month before we go to print. And, as usual, the absence of an impending trip to the printer has failed to fill me with the fear necessary to inspire me to write one of my usual brilliant diatribes. So, this time, I think I'll describe exactly how we put together this fabulous periodical that we like to call "*GS+* Magazine".

Week 1

The activities of the first week of work on an issue of *GS+* Magazine are actually filled with work on the *previous* issue. This is time that we spend duplicating disks for the previous issue and buying last minute supplies (mailing labels and envelopes) that we may have overlooked in the previous weeks. If we get lucky, the magazine will come back from the printer during this time, and we will get to mail out all of the magazine-only subscriptions by late Friday.

The work we do on the *next* issue is mostly idea work. What will we review? Who will review it? What programs will Joe and I work on? Which new advertisers will we try to get? All of this brainstorming usually gives us five or ten solid pages of material for the next issue.

Week 2

This week is, once again, filled with work on the previous issue of the magazine. The magazines usually arrive back from the printer late Friday of Week 1 or early on Monday of Week 2, so this is the week that we spend mailing them out. However, for the most part, Joe spends all of his time working on programs, and the rest of us spend the days doing what we affectionately call "gripping and sticking." This is the process of inserting the disks into the little baggies, removing the backing material from the baggies, and sticking the baggies into the magazines. This is also the time we prepare all of the First Class and Foreign subscriptions. This also involves a variation on gripping and sticking: the act of placing magazines and disks into the mailing envelopes, and sealing the envelopes.

Not much brainstorming or idea work is actually done in week two—our main focus is getting the magazines mailed out. However, as I said earlier, Joe spends week two working on his programs for the next issue. This usually means that, by Wednesday of week two, Joe has determined that he doesn't actually want to work on *that* program, and he starts working on something entirely different. By Friday, he's decided that the first program wasn't actually so much trouble, and he's almost finished with it.

Week 3

Joe is still programming during Week 3, but for me, this is catch up week. I have to get back on track answering all of the mail and taking care of all the expenses and bills that got ignored during weeks 1 and 2. About this time, Michelle starts asking Joe and me where all of our articles are. Being veterans of this sort of thing, Joe and I usually just say, "I didn't know *I* was supposed to work on *that*!" and return to what we were doing.

This is also the week when the phone starts to ring a lot. On the other end are concerned subscribers asking "Where's my magazine?" The usual answer is, of course, "It's in the mail." If you live in California, the answer is, "We sent them out last week, but with that earthquake (or flood, or fire, or mass Elvis sighting) you guys just had, it could add a few days to the travel time." If you live in the North Eastern United States, the answer is, "We sent them out last week, but with all that snow and ice you guys have gotten, it may take a while for you to get yours." Surprisingly, this answer works all year.

Week 4

This is panic week. We still have only five or ten pages of material decided on—and one of those is the table of contents! So, we brainstorm some more, and we come up with a few more pages. If we are lucky, we also come up with a concept for the cover photo. This is also the week that I usually write this column.

During Week 4, Joe and I give all of the small stuff that we've finished (like this column, and maybe the "Glossary" department) to Michelle so that she can lay it out. When she finishes an hour later, she starts looking at her list of blank pages and mumbling to herself.

The rest of this week is spent avoiding Michelle.

Week 5

If we were able to come up with a cover concept during Week 4, this is when Bob shoots the photo for it. He also shoots the photos of any hardware that we are going to be reviewing. (In case you are wondering what else Bob does around here . . . Bob is the backbone of the organization. While the rest of us try to write reviews and programs, Bob is here every day handling the incoming orders. He also drinks his own weight in milk each week, and constantly complains that we should publish more games.)

During this week, things actually start to get written! Mostly, these are reviews that I've put off for as long as possible, so that I can work with the products enough to give them a thorough review.

As these begin to trickle in, Michelle and I edit them, and she gets ready to lay them out during Week 6. At this point, things start to look pretty good. The length of the material that we have begins to expand past our original estimates, and we now have about 25 pages of material. Only 25 more to go!

Week 6

The is the week that Joe starts putting together the contents of the disk. You can tell when he is finished because he says either, "We have 500K free," or "We need to chop about a meg from the disk."

Also about this time, I start to finish up whatever programming project I've been working on, unless we need to chop stuff to fit everything on the disk. In that case, I try to finish up any reviews or articles that I have left to do. During this time, Michelle is busy doing layouts, and contacting advertisers.

Week 7

This is the week that Joe and I finish up all of our articles and programs and turn them into Michelle. She begins the final layouts of all these and then, on Friday, she announces, "We have 20 blank pages, none of the ads have arrived yet, and we go to the printer in a week. Thank you."

Week 8

This is the week during which I find a kind of inner peace. The phone is already ringing with people asking where their magazines are, Michelle is constantly threatening us with all sorts of bodily harm, and none of the ads have arrived yet. Still, there is peace in knowing that, with luck, I'll have a fatal heart attack, and I won't ever have to do this again.

And then, a miracle occurs, and early Friday morning, we take all of the pages to the printer!

Then we do it all over again.

Diz

CONTENTS

FEATURE ARTICLE

PROGRAMS

DEPARTMENTS

REVIEWS

Products marked with a bullet (•) were review copies provided by the publisher.

ADVERTISERS

GS+ Magazine

March-April 1994
Volume 5, Number 4

Publisher, Editor
NOREEN M. "TAXMAN" DISBROW

Technical Editor
JOSEF W. "JOLLY ROGER" WANKERL

Production/Marketing Coordinator
MICHELLE L. "BUXOM WENCH" RIBARIC

Operations Director
ROBERT A. "BUCCANER BOB" RIBARIC

Lowly Contributing Editor
STEVEN W. "CABIN BOY" DISBROW

On the Cover

MUTINY!!!!!!!!

We have had enough!!!

Special Appearences by Nesbit and Ahzeibah

(April Fools', Steve)

Subscription rates - Magazine only:
1/2 year (3 issues) - $10
1 year (6 issues) - $18

Subscription rates - Magazine w/Disk:
1/2 year (3 issues) - $20
1 year (6 issues) - $36

Tennessee residents add 7.75% sales tax.
Add $2.00 per issue if you want First-Class delivery.
Canadian and Mexican orders add $2.00 per issue.
Other foreign orders add $2.00 per issue for surface delivery or $5 per issue for Air Mail.

Send orders, ads, inquiries, and address changes to:
GS+ Magazine
P.O. Box 15366
Chattanooga, TN 37415-0366

Or phone Monday-Friday 9 am - 5 pm Eastern Time:
(800) 662-3634 (orders only)
(615) 843-3988 (tech support)
(615) 843-3986 (fax)

If you have a submission for *GS+* Magazine, send it to:
GS+ Submissions
P.O. Box 15366
Chattanooga, TN 37415-0366

GS+ Magazine can also be contacted on these online services:
America Online: send mail to GSPlusDiz
or visit our online area by using the keyword GSMAG.
Delphi: GSPlusDiz
GEnie: JWankerl
InterNet: gsplusdiz@aol.com

GS+ Magazine is published bimonthly by:
EGO Systems
7918 Cove Ridge Road
Hixson, TN 37343-1808
(DO *NOT* SEND MAIL TO THIS ADDRESS—USE FOR UPS AND OVERNIGHT DELIVERIES ONLY!)

GS+ Magazine is produced on the Apple IIGS using the following products:
EGOed - Text Editing
GraphicWriter III - Interior Page Layouts
AppleWorks GS - Cover Layouts
Platinum Paint - Screen Shots
Proof pages are printed on an Apple LaserWriter IINT.

Letters

[Editors note: On the Feedback forms we've been getting, lots of you have been requesting an expanded "Letters" section. Therefore, since so many of you were kind enough to write us, we're giving you four full pages of letters in this issue! Let us know what you think! - Diz]

Dear Diz:

Delighted to receive your always informative and interesting magazine. Two points:

1. TypeSet was reviewed in the Sept/Oct '93 issue, but no mention was made of the nasty bug involving the Key Equivalent function. It simply doesn't work! I made WestCode aware of that last September at which time they acknowledged the problem and promised a fix, which, until now, has not been delivered. Could you follow up on that one?

2. The CD-ROaM article in the Nov/Dec 93 issue was interesting, but some of us are still in the dark about:
 a) Which ROM device is suitable (without CD-ROaM) for use with an Apple IIGS and
 b) Which software is currently available.

It would be helpful to have a follow-up article on those two points.

Henry F. Sherwood
Hossegor, France

The Key Equivalent option in TypeSet does *work, it just doesn't work* immediately. *After setting a key equivalent, you have to switch programs or reboot in order for TypeSet to use the new key equivalent. Sorry about that, but Joe and I couldn't figure any other way to do it.*

As for CD-ROaM, there is basically one factor that determines what type of CD-ROM drive you can use with your IIGS. That is: what kind of SCSI card do you have?

If you have a RamFAST/SCSI card, you can use just about any *CD-ROM drive that your heart desires. If you stick with a fairly well known brand (NEC for example), you can't go wrong. However, if you have any doubts, you should contact Sequential Systems (1200 Diamond Circle, Lafayette, CO 80026, or call 303-666-4549) for complete compatibility information.*

If, however, you have an Apple II High-Speed SCSI card, you are severely restricted in the kind of CD-ROM drive you can use. Basically, you have to stick with either an older Apple brand drive, or an older NEC drive. The old AppleCD 150 drive works great (and you should be able to find them dirt cheap—for example, MacWAREHOUSE [800-255-6227] is selling the old AppleCD 150 for only $149.95 in their current catalog), and I personally use an old NEC CDR-36 on my IIGS. However, to use the NEC drive, I have to also use the CD-ROM drivers that are sold by Tulin Technology (2156H O'Toole Ave, San Jose, CA 95131, or call 408-432-9057), so that the Apple SCSI card can recognize the drive. For more information on how all of this actually works, you should check out my article "The Scavenger" in GS+ V4.N5.

As for what CD-ROM software is available, that was the reason I wrote CD-ROaM . . . there really isn't anything out there that is specifically *for the IIGS. There are two CD-ROMs intended for Apple II owners: the 1990 GEM Apple II CD-ROM [see review in GS+ V4.N2], and the System 6.0 Golden Master CD-ROM. Beyond that though, we IIGS owners basically have to resort to scavenging files off of CD-ROMs intended for other platforms. For example, I reviewed the Key Fonts Pro CD-ROM in GS+ V4.N5. This CD-ROM contains over 300 TrueType fonts, and was intended for purchase by Macintosh and Windows users. However, since the disk is formatted for the Macintosh, it works great on the IIGS using System 6 (or later), the HFS FST, and Pointless. (For a complete example of how this works, see "The Scavenger" in GS+ V4.N5.)*

And then there's discQuest from Sequential Systems [see the next letter], which promises to make dozens of new CD-ROM titles accessible on the IIGS. At this point, I still have not seen discQuest, so I can't tell you how well it actually does what it does. However, we will be reviewing it as soon as we can.

Diz

GS+:

I just wanted to drop a line concerning the November/December [1993] issue.

1. Balloon is an excellent Finder extension! This is the type of program for which I subscribe to *GS+*. If you plan an update, it would be nice to be able to extract only selected files from an archive. Of course, a Finder extension to create NuFX archives would be ideal (e.g., select one or more files or folders and select an item from the Extras menu, and a NuFX archive is created). This would possibly place you in competition with AutoArk and HardPressed, so you would need to weigh the possible advertising losses before doing this. I do not own either of those programs, but, I own StuffIt Deluxe and StuffIt SpaceSaver for the Macintosh. [On my Mac,] I find that I use manual compression from the Finder (using Balloon-like utilities) more than I use automatic compression/expansion.

2. I would like to see a review of discQuest. It appears from my reading that this is a DiscPassage clone. (DiscPassage is a program for Macs and PC that is used to read specially formatted data from CD-ROMs.) I own several of the listed compatible CD's and they are all DiscPassage based disks. If this is true, then discQuest should be compatible with future DiskPassage-based CD's. . . .

I have been thinking about buying a faster CD-ROM reader for my Macintosh and giving my old one to my IIGS, but have been hesitant to do this in the past because there were essentially no CD-ROM's for the IIGS. I hope that discQuest sells well enough to stimulate development of IIGS-specific disks. The IIGS is a capable multimedia machine, but real, useful multimedia requires a lot of sounds, graphics, and animations [translation: megabytes], and current IIGS publications such as Script-Central and Stack-Central are limited by the 800K floppy format. By the way, I don't think that diskQuest will support any multimedia encyclopaedia (Compton's and Grolier's use their own interfaces). Unfortunately, educational users (school's with IIGS's) would probably find the most use for an encyclopaedia.

In conclusion, the November/December issue was the best issue in the last years. Keep up the good work.

DuBose Medlock
via America Online

Glad that you liked Balloon! Joe is working on an update, but we aren't sure when it will be published. The reason is that we will have to secure another license for the LZW methodology in order

to publish another version of Balloon. (The licence that we got for Balloon v1.0 doesn't allow us to publish any updates.)

Diz

Dear Steve,

. . . I need your advice on choosing a second printer for my IIGS. The ImageWriter II is great, but its text printouts are nothing to be proud of.

. . . The printer that I want is an inkjet which meets the following requirements:

(a) Able to produce high quality printouts from text-based programs like AppleWorks 3.0.
(b) Able to print with 16-bit graphic-based programs like Platinum Paint 2.0, HyperStudio 3.1, and GraphicWriter III (and of course, Print Shop GS).
(c) Have at least a choice of three resident fonts, one of which should be proportional.
(d) Not too big and bulky like the HP DeskJet 500.
(e) Can use all kinds of paper. (Some inkjets use special paper only).
(f) Easy to configure and operate.
(g) Supports an Epson emulation printer driver. . . .

. . . I also need your advice on how to connect the new printer and my existing ImageWriter II to the printer port without having to unplug one and plug in the other whenever I want to use one of them. I understand there is a switch box that allows you to do this. Is that so? . . .

. . . What I have in mind is one of the latest inkjets on the market—the Canon BJ-200. . . What do you think? Have I made the right choice? . . . .

Zainudin Hashim
Kuala Lumpur, Malaysia

First of all, I have to confess that I have absolutely no experience with the BJ-200 printer, and that the following advice is mainly based on what I know about other *products and their use on the IIGS.*

So, the short answer is "no." I don't think the Canon BJ-200 is the right choice. The main reason is that there isn't a IIGS-specific driver available for this printer. (At least, not one that I'm aware of.) So, while you might be able to use the printer in Epson emulation mode, you wouldn't be able to use it for much more than that.

Actually, I would strongly recommend that you reconsider choosing a printer from the HP DeskJet or DeskWriter family. These printers fit all of your criteria except for item "d", and there is a new portable DeskJet (similar in appearance to the StyleWriter) that will probably solve that problem.

Another nice thing about the DeskJet is that there are two different brands of IIGS-specific drivers available for them (Harmonie and Independence) that are full-featured and produce excellent output. (For more information on Harmonie and the DeskWriter 550C printer, see the review of the DeskWriter 550C in our last issue.)

As for connecting the printer to your IIGS, you can use a simple serial A-B switch box to hook both your ImageWriter II and DeskJet at the same time. Before I networked everything, that's how I hooked my old ImageWriter II and DeskJet together and it worked great. These switch boxes should be available at your local computer or office supply store for around $20. (Note that most Hewlett-Packard printers have both a parallel and serial interface, so you won't have to buy a parallel interface card to use it with your IIGS.)

Finally, if the only reason you are getting a new printer is because of the text output of the ImageWriter II, you should get yourself a copy of Pointless before you spend all your money on that new hardware. The output that you get from Pointless and your ImageWriter II might just change your plans.

Diz

GS+:

I just read the letter to the editor (in *GS+* V5.N3) from the gentleman who wanted to know if there was any way to transfer PaintWorks Gold pictures to Print Shop IIGS format. I do know that there are a couple of ways.

One way is to use the Print Shop Companion IIGS. This is a standalone extension to Print Shop IIGS, and has several nifty functions . . . graphics editor, calendar maker, cataloger, and so on.

To use this option, save the PaintWorks Gold picture in "screen" format. Actually, it's best to shrink down the center of interest before saving because when you use the import command of Print Shop Companion's graphic editor, you have to mark out just a small portion of the picture to be converted. It will also convert single and Double Hi-Res pictures, but it can't deal with Apple Preferred Format pictures.

One more caveat . . . Print Shop Companion doesn't seem to like System 6.0.1, at least not on my machine. I have to start it up from the floppy. If you do a lot of Print Shop stuff, though, it can be very useful. It even supports the New Print Shop format.

The other option is The Graphics Exchange, from Roger Wagner Publishing. This program allows you to convert graphics to and from all sorts of Apple II formats. You can change a file from Super Hi-Res to Print Shop IIGS or Double Hi-Res or even Low resolution graphics (the program does it's best to retain as much quality as possible , but of course, you do lose something in the translation, especially when going to a lower quality format). It's especially good if you're just moving up to the IIGS from another Apple II . . . you can change those old Hi-Res or Double Hi-Res pictures into Super Hi-Res, then enhance them.

The program is a little hard to use, but it does have a lot more flexibility in terms of the formats it supports. Also, you can choose whether you want to mark out a section of the screen to convert, or whether it should convert the whole picture.

The current version is desktop based, and supports all of the desktop features (NDA's, FST's CDA's, etc). Personally, I only use it when I have to, because the interface is kind of hard to navigate, but it does do some things that no other program will.

So which one is best? It's hard to say, as they're basically two very different programs with one very small area of overlap (transferring Super High-Res graphics to Print Shop IIGS format). If you're mainly interested in doing Print Shop stuff, go with Print Shop Companion. If you're interested in moving graphics between all sorts of Apple II formats, go with The Graphics Exchange. If you do both, get 'em both.

Edward O'Hara
via America Online

Thanks for the help Ed! As you requested elsewhere in your letter, I looked up the pricing for both of these items: Print Shop Companion IIGS is $29.95 from Quality Computers (20200 Nine Mile Rd, St. Clair Shores, MI 48080, or call 800-777-3642), and The Graphics Exchange is $49.95 from Roger Wagner Publishing

(1050 Pioneer Way, Suite P, El Cajon, CA 92020, or call 619-442-0522).

Diz

GS+:

Just a note in appreciation of your great magazine and the programs you offer. In [V5.N2], you had a game called Kablooie! that came with a warning about it being a bit addicting. I guess you might call it addicting!

. . . On a separate topic, I have to use this space in support of the most important IIGS product of the past year, GNO v2.0. I am not sure if you have done a review of GNO (maybe 1.0) but 2.0 really deserves some real attention of non-programmers. Right now, I am in Kablooie! where I checked my scores, and I am dialed into my school's UNIX system composing this letter. If only I could put Kablooie! down long enough to develop more GNO utilities . . . .

Jeremy Rand
via the InterNet

Glad you like Kablooie! Jeremy! We've gotten lots of letters like your telling us that they not only like Kablooie!, but that they want to see more games. All I can say is, don't expect too many games in GS+ Magazine, but we will definitely be publishing more than we have in the last four years.

As for GNO, I hope you saw our review of it in the last issue. If not, check it out and let us GNO what you thought of it.

Diz

Dear *GS+*,

Well, I am catching up on some of my reading, I have finally gotten up to the October [1993] issue, and I ran across Bill's article, "So You Bought a Hard Disk . . . Now What?", and as I read through it, I keep on spotting two things, "HFS" and "Backup." These words and a problem that I had when I first made an HFS partition on my hard disk, brought an idea to my mind. Let's suppose you accidentally delete something on your HFS partition, or better yet, you have some major problem with the file structure on the HFS partition. What do you use to fix it? ProSel-16 won't fix it, and neither will the Salvation utilities (at least not the versions I have), but you mentioned Universe Master. Will Universe Master diagnose and repair problems on an HFS partition? It is because I have no way to recover from these problems that I no longer have an HFS partition on my hard drive. If you can tell me how to fix these problems on an HFS partition, you will have made my day. . . .

The next thing . . . is the review of Prism by [Bill Moore.] [This review appeared in *GS+* V5.N2 - Ed.] I think that [Bill] has done a great job with this article, and has indicated every shortcoming of Prism that I was able to find. But when comparing it to SuperConvert v3.1, I think he fails to mention some of the other strong points that Prism has over SuperConvert. [For example,] I have failed to find a GIF of any size (non-interlaced of course) that will not convert properly with Prism. SuperConvert has a big problem in that it will not handle files much larger than 640x400, if that. Not that it is that big a deal, considering that AppleWorks GS doesn't like to import graphic files much larger than screen size. Prism does a better job of converting 320 to 640 gray scale . . . .

I agree that they do make good complements to each other, but they both lack a variety of features, and have little quirks that make them irritating . . . .

Charles J. Sarchet
Mare Island, CA

Sadly Charles, I don't think that there are any IIGS utilities to recover files from a HFS disk. At this point, about all any of us can do is back up our data regularly. Then, if disaster strikes, you can just re-format the disk and restore the data. I admit that it isn't the best solution in the world, but it works.

Diz

Dear *GS+*:

Why are there new **Icons** folders created each time you launch [a newer application] from a partition other than the System partition? I have several partitions and now each seems to want to have its own **Icons** folder.

Is there anyway to get the new type icons that reside in the invisible **Desktop** file, that is inside the **Icons** folder, to all migrate to the system **Icons** folder where they belong?

Rusty Schneeflock
St. Joseph MO

Well Rusty, whenever you run an application, the Finder looks inside the application to see if there are any icons associated with it. If the Finder finds any icons, and the Finder doesn't already have copies of them, it then copies them out of the application, and puts them in the ***Desktop*** *file of the disk that the application is on. If there isn't a* ***Desktop*** *file, then the Finder creates an* ***Icons*** *folder on the disk, and then the Finder creates the* ***Desktop*** *file in the new* ***Icons*** *folder.*

The reason that it does this, as opposed to putting everything in the ***Icons*** *folder of your boot disk, is known only to the guys that wrote the Finder. However, I think that if I were writing the Finder, I probably would have done it the same way. Why? Because, if you kept all of your icons on your boot disk, it would quickly become a huge file. This would rapidly become very bad news for people that have to boot their IIGS from a floppy disk.*

As for copying icons from one ***Desktop*** *file to another, at this point, I don't think that there is any easy way to do that. If you really want to do it though, you could delete all of the* ***Desktop*** *files on your other disks, copy all of your applications to your boot disk, and then run them from there. The first time you run them, the Finder would add their icons to the* ***Desktop*** *file on the book disk. After that, you could copy your applications back to where you want to normally run them from.*

However, since that would be a major pain, maybe someone out there will see your letter and write an application to do just that.

Anyway, for more information on how icons work, see my article "Casual 6 - Those Crazy Icons*" in* GS+ *V4.N2.*

Diz

Dear *GS+* Magazine:

. . . I want to ask you a question that I hope you'll have the answer to. I'm using a HP 550C [printer] and am pretty happy with it. However, the industry is moving on to 600 d.p.i. [printers]. I know that the IIGS Print Manager makes this step up non-existent *but*, is it possible to create a PostScript file with the LaserWriter printer driver, send it to the printer and have the printer do all the work? I'm assuming that there would have to be a utility program created for the IIGS, similar to the Mac's LaserWriter Utility program. I have tried printing out [on a Macintosh,] a PostScript file which I created [on the IIGS] . . . . The layout printed out correctly, but not all of the fonts printed correctly . . . . I'm

assuming that this is mostly due to TrueType fonts being downloaded and used versus Type 1 [PostScript] fonts. Perhaps the new Apple LaserWriter Select 360 printer would solve the TrueType problem since it has a built in TrueType rasterizer.

Mike Wallace
St. Clair Shores, MI

Well, first and foremost, you are proceeding from a false assumption. The IIGS can use a 600 d.p.i. PostScript printer just as easily as it uses a 300 d.p.i. PostScript printer. The key word here is "PostScript." The reason is that a PostScript device does not know the difference between 300 d.p.i., 600 d.p.i. or 15 million d.p.i! All it knows is that somebody sent it a print job, and it needs to print that job just as soon as it possibly can.

Having said that, I need to point out that there is a IIGS equivalent of the Macintosh LaserWriter Utility. It's called LASERbeam, and the latest version of it is in this very issue of GS+ *Magazine. (How's that for fast service?) You can use LASERbeam not only to download PostScript files, you can also use it to download Type 1 PostScript fonts to your favorite PostScript printer. Best of all, if you have the bit-mapped or TrueType equivalent of the Type 1 fonts you download, you can use these fonts in your IIGS documents and print them out on your PostScript printer at the full resolution of the printer!*

Now, let's look at why your IIGS-created PostScript file did not print out correctly from the Macintosh. When the IIGS LaserWriter driver creates a PostScript file, it looks at the names of the fonts that are used in your document, and converts these names into "PostScript-style" font names. These PostScript-style names tell the printer exactly which fonts to use when printing your document. If the name that the LaserWriter driver generates for a font doesn't match exactly (case included) the name of one of the fonts in the printer, the LaserWriter will not find that font. The result will be either a missing font, or the use of a default font that doesn't look very good. Unfortunately, the IIGS LaserWriter driver uses a convoluted process for creating PostScript-style font names. The result of this process is that these names rarely, if ever, match the names of the fonts download to printers. That's why some of your fonts came out incorrectly: The names of those fonts in the PostScript file did not match the names of the fonts in the printer.

The solution to this problem is to create your documents using only fonts that the IIGS LaserWriter driver knows about (and can therefore generate the appropriate names for [see IIGS Technical Note #67, which is on your GS+ *Disk, for a list of the fonts the IIGS LaserWriter driver knows about]), or, use LASERbeam to download the Type 1 PostScript fonts with the appropriate names (LASERbeam automatically renames fonts the same way the IIGS LaserWriter driver does), and then print the document from your IIGS. (If you are still confused about all of this font naming business, there is a complete discussion in the "LASERbeam" article, elsewhere in this issue.)*

Finally, let's talk about TrueType fonts and printers with TrueType rasterizers. Basically, all a TrueType rasterizer is, is a PostScript program that knows how to interpret TrueType fonts and render them on a PostScript device. So, while a printer may have a TrueType rasterizer built into it, and the IIGS can use TrueType fonts via Pointless, it all means absolutely nothing without some way to actually download those TrueType fonts to the printer and let the TrueType rasterizer know that they are there. At this point, there isn't a IIGS utility (not even our own LASERbeam) that can do this.

Diz

Dear *GS+*:

I have looked everywhere in my town for a 3.5-inch disk mailer, but I can't find one. If you have some, I will buy some from you. I would like to get the System 6.0.1 upgrade from you, but your instructions say I need to send a disk mailer! . . . Any help would be appreciated.

James Hissong
Eglin AFB, FL

Well James, you don't actually need an official "disk mailer." Any kind of padded envelope that is big enough to hold five disks will work just fine. In fact, your local Post Office sells several different sizes of padded envelopes that will work. Just drop by or give them a call to see what sizes they have and what their current prices are.

Diz

AKKK!
I use EGOed for all of my short word processing, but A2-Central [is] moving more and more to a compressed format. Can we ever cope without getting EGOed to read these files directly?

Will I actually have to leave the Finder?

Could I find my way back?

I will wait with bated (or is that baited) breath. For your reply.

(Ignoring those files was a cop-out, load 'em up and save them uncompressed.)

Name Withheld

Even though this is our annual April Fools issue, I must point out that this is an actual letter that we got recently. (I admit that I did touch up the punctuation and capitalization a bit, just so it wouldn't be too *silly.) Now, I love to get letters from our subscribers, and I really do read every single one of them, but to be honest, I really don't have a clue as to what the author of this letter is talking about. (This is just a guess, but I think he is mad at us because EGOed v2.0 can't read some file format that is used by* A2-Central *On Disk.)*

So, I'm running this letter to make a point: When you write to us, please *take the time to write coherently! Please explain* exactly *what it is that you want to say, and always try to include a beginning, a middle, and an end in your letters. I would rather spend 10 minutes reading a detailed and coherent letter once (and then being able to answer it immediately), than spending that same 10 minutes (or longer) trying to figure out exactly what the heck the author was talking about.*

Diz

If you have a question, comment, or criticism about *GS+* Magazine, we want to hear it! Due to space limitations, we cannot answer every letter here in *GS+* Magazine.

If you want a personal reply, please include a daytime phone number, or enclose a self-addressed, stamped envelope with your letter.

Please address all letters to:

GS+ Letters
P. O. Box 15366
Chattanooga, TN 37415-0366 *GS+*

We're proud to announce the... PROCYON Spring '94 Product Rollout!

Who are we, anyway?

Procyon was founded on the princple that a person doesn't have to buy the latest whizz-bang hardware technology and play the obsolescence game just to get work done on their computer. For the past two years we've been working towards our dream of producing useful and high-quality software for the Apple IIGS (and making money doing it), and now with the introduction of these products our dream has finally come true. See for yourself - the Apple IIGS, regardless of what the local computer salesmen is likely to tell you, is not dead.; it is at this very moment undergoing a revitalization as dedicated companies like PROCYON help fulfill the potential of the Apple IIGS microcomputer.

GNO/ME™ 2.0

The newest version of our amazing UNIX™ implementation for the Apple IIGS. GNO 2.0 includes an automatic installer, bug fixes, ORCA/2.0 compatibility, remote (dialup) access package with password protected logins, rewritten and complete documentation, standard communication and file transfer utilities (ZModem), and much, much more! GNO/ME is also completely compatible with the ORCA and APW tools and languages, so your work can immediately benefit from GNO's multitasking power. Recompile your program in the background while editing source code, viewing ShrinkIt archives, or writing a letter to mom! Built-in communications facilities make talking to other processes (or other computers) easier and more powerful than ever before! Check it out!

Splat!™ 1.0

The most powerful source-level debugger yet for the Apple IIGS programmer (and ORCA™ tools) is also the easiest to use! Splat! utilizes a text-based "desktop" environment so your choices are always visible. It supports ORCA/C 2.0, ORCA/Pascal 2.0, and any other language that uses the ORCA standard debugging format. Splat! has full support for structures and arrays that no other debugger in its class does, including the ability to modify any element in them. Profiling allows you to to find the slow parts in your program and concentrate on optimizing just those portions. If you're just learning to program, Splat! is the tool you need to help you find "beginner's bugs" - the errors that can make learning programming frustrating.

Splat! is the kind of tool programmers on other platforms have had for years - so get Splat! and see what you've been missing!

Procyon provides unparalleled technical support via phone and all major online services:
GEnie: PROCYON.INC, A2Pro RoundTable
AOL: GNO Jawaid, ADV Forum
Delphi: JAWAIDB
Internet: bazyar@cs.uiuc.edu, comp.sys.apple2.* newsgroups,

For a complete catalog of Procyon products, call or write:

Procyon Enterprises Incorporated
Apple IIGS Software Publishing
P.O. Box 620334
Littleton, CO 80162-0334
(303) 781-3273

Programming the IIGS - Part 1: Getting Started

By Steven W. Disbrow

Over the years, we've gotten a lot of calls and mail on a lot of different subjects. However, the one subject that lots of people want to discuss is programming the IIGS. More specifically, they want to know how to get started on the road to programming the IIGS. So, that's what this series of articles will be about: Programming the IIGS.

In these articles, I'm going to assume that "Programming the IIGS" means writing IIGS-specific, desktop-based programs. Therefore, I won't be discussing AppleSoft BASIC, older models of the Apple II, or anything else that won't let you take full advantage of the capabilities of the IIGS.

Getting Started

This first article is intended to be a "packing list" for beginning IIGS programmers. In other words, this article is an absolute rock bottom, ground zero introduction to writing programs on the Apple IIGS. This article will *not* discuss any programming techniques or methodologies. All this article is intended to do is to tell you *what* you need to get started programming the IIGS, and how to get in touch with the vendors that provide the products and services you need. It also discusses *why* you need to get these things before you start programming the IIGS.

Experience

The first thing you need to program the Apple IIGS is, actually, some previous programming experience. While this may sound like I'm coping out here, I'm not really—after all, this series is about programming the *Apple IIGS*, not programming in general. And, like it or not, if you don't have some previous programming experience, you are going to have a very hard time trying to program the IIGS.

So, how do you tell if you have enough programming experience? Well, for starters, ask yourself the following questions: Do you know what a "record" is? Do you know what a "pointer" is? Do you know what a "bubble sort" is? Do you know what the phrase "compile and link" means?

If you answered "no" to any of those questions, you probably need to get a little more programming experience under your belt before you tackle writing a IIGS program. To do that, you either need to take a beginning programming course at the school of your choice, or you need to get yourself a copy of the Byte Works Learn to Program in C (reviewed in *GS+* V3.N3) or Learn to Program in Pascal (reviewed in *GS+* V3.N4). These are self-paced courses that are intended to teach the basics of programming to the novice.

Note that while these courses work *with* the IIGS, they don't focus on programming the IIGS—their main focus is on teaching the basics of programming. If you are a true beginner, this is exactly the type of course you need. (Note that, for the beginner, I would recommend the Pascal course. This is simply because Pascal is a better language for beginners to learn with. We will discuss this in more detail later on.)

More Memory & A Hard Disk

These are two items that we've been pushing for all IIGS owners since the day we started publishing *GS+* Magazine. But, where these items merely make the IIGS easier to use for the average user, they are indispensable for a programmer. Without adequate memory, the programs you write might not even be able to compile and link (if you don't know what "compile and link" means, trust me, it's *very* important, and not being able to do it is *very* bad). Without a hard disk, you're programming time will be filled with tedious disk swapping, which isn't a great productivity booster.

So, if you really want to program, get yourself as much memory and the biggest hard drive that you can afford. Even if you don't end up as a IIGS programmer, having a IIGS with more memory and a hard drive will make your life a lot easier!

Books, Books, Books

Next, you need to pick up a few books. In specific, you need to get all of the books shown in Figure 1. While it will make your life much easier if you have access to all of these books, you can actually get by with just the titles that I've underlined. Having said that, be sure to read over all of the information in Figure 1 so that you know what each book contains and why you need it.

Looking at that list, you may be starting to panic a little. Don't worry, you don't have to read all of these books at once. In practice, you'll only use these books when you need to look up a tool call that you've never used before, or when you want to verify something about how the IIGS works. In other words, these are *reference* books, not tutorials, and you can't use them to learn how to program the IIGS. However, if you try to program the IIGS without these books, you will be fighting a losing battle.

Stuff That Helps

Now, in no particular order, here are some other things that you should try to do before you start your IIGS programming career:

• Get some sample source code: Books and tutorials are great, but there is no better programming learning aid than a completely functional program that you have the source code to. By looking at the source code to a functional program, you can see how someone else tackled a certain problem, and you know that it works. If you already subscribe to *GS+* Magazine with the disk, you are pretty much covered on this one. If you don't, you should consider it—every program that comes on the *GS+* Disk comes complete with source code. (I'll point out other sources for source code as they come up.)

• Get an online account: If you don't have a modem, you should get one and get online as soon as possible. Being online gives you access to fellow IIGS programmers around the country (or even around the world), that have already gone through exactly what you are going through: learning to program the IIGS. If you ask nicely, you should be able to find a whole community of folks willing to help you in your programming endeavors.

At this point in time, GEnie seems to be the online service of choice for most IIGS programmers. But, at the end of this article, I'll give you contact information for all of the services that I personally use. (By the way, the online community is not just a great place to get advice, it's also a great place to advertise your new whiz-bang program when you finish it. In fact, *GS+* Magazine started with just a single advertisement that I placed in the classifieds board of America Online.)

• Buy more books: After you have all of the IIGS technical manuals that I discussed earlier, you will still need to buy more books that are related to the program you are trying to write. In other words, if you are going to write a program that controls the LaserWriter, you need to get the *LaserWriter Reference* book from Apple. However, purchasing this type of book can wait until you actually decide on the kind of program you want to write. (I know, this *sounds* obvious, doesn't it?)

• Join a user group: Besides giving you a warm fuzzy feeling inside, joining a user group is a great way to meet other *local* IIGS programmers. This can lead to more sample source code, late night programming sessions, and a new friend or two.

Some Other Resources

Just in case all of the above weren't enough, here are some other IIGS programming resources that you should keep your eyes open for:

• Technical and File Type Notes. Technical Notes are documents, published by Apple, that detail bug fixes, work-arounds, and other things that you might need to know about for your IIGS programming projects. File Type Notes are documents that give detailed information on the files that are used by various Apple II and IIGS applications. Together, these notes fill in many of the gaps that exist in the books listed in Figure 1. The question then is, "How do you get them?" Well, the easiest way to get Technical and File Type Notes is to subscribe to one of the online services listed in the contact information at the end

Figure 1
Books You Need, and What's In Them

Apple IIGS Toolbox Reference: Volume 1, Volume 2 and Volume 3 - If you don't have these three books, you don't have a prayer of writing a decent IIGS program. These books contain all the details about using the hundreds of tools that are built into the IIGS and its System Software. These tools are what you use in your programs to give them the distinctive look and feel of a IIGS desktop application. The first two volumes detail all of the tools that were available (or were soon to be available) when the IIGS was introduced. The third volume details all of the new tools that were introduced with System Software v5.0. As a set, these books detail every tool that was available in the IIGS Toolbox up until System Software v6.0 was introduced.

Programmer's Reference For System 6 and *Programmer's Reference For System 6.0.1* - When System 6 was finally released, Apple Computer Inc. was pretty much out of the business of supporting the Apple IIGS. So, it fell to the Byte Works to publish these two volumes detailing the changes, bug fixes and additions to the Apple IIGS Toolbox that came in System 6 and 6.0.1. Like the *Apple IIGS Toolbox Reference: Volume 3*, these books detail new tools that were added to the IIGS Toolbox with the release of each new version of the System Software. In other words, you can write IIGS software without these books, but you won't be able to take advantage of all the neat new features in System 6 and System 6.0.1.

Apple IIGS GS/OS Reference - This book will tell you everything you need to know about the Apple IIGS operating system (GS/OS) and how to use it in your programs. (This book also discusses the old ProDOS 16 operating system, so if you have this book, you do *not* need the *Apple IIGS ProDOS 16 Reference*. So, if someone trys to sell you that book as a substitute for the *GS/OS Reference*, don't buy it.) Other topics covered include File System Translators (FSTs), the Apple IIGS Loader, GS/OS error codes, and the Apple IIGS object module format. If the program you are going to be working on has to do any file manipulations at all, you need to get this book.

Human Interface Guidelines: The Apple Desktop Interface - If you are going to write a IIGS desktop program, you need to read this book. It tells you what need to know to create an application that follows Apple's guidelines for interacting with the user. By following these guidelines, your application will be easier for users to learn to use. Ease of use makes users happy, and happy users are more likely to be repeat customers. (Note that a new edition of this book has been published in a new format and with a slightly different name: *Macintosh Human Interface Guidelines*.)

of this article. As these notes are updated, Apple posts the latest versions to all of the major services.

• Back issues of *8/16-Central* and *Nibble* magazine. Although these also focused quite a bit on the older 8-bit Apple IIs, both of these magazines published lots of IIGS technical articles and source code.

• Back issues of *GS+* Magazine: Not to toot our own horn, but *GS+* Magazine has focused on programming the IIGS from day one. Just about any back issue that you get will have some sort of programming article in it, and all of our back issue disks have source code on them. Best of all, it's all IIGS-specific. In particular, you should try to obtain as many of the "Working With the Toolbox" articles as you can. These articles discuss individual IIGS tool sets, and have companion programs on the *GS+* Disk that show you how to use these tools from a real program.

A Programming Language

Finally, after you have all of the above stuff squared away, you need to choose the programming language that you want to use as you work on the IIGS. You may be wondering why I saved this for last. The main reason is that your choice of language might be influenced by your ability (or inability) to obtain some of the above items. For example, if you can't find anything but Pascal language sample source code, you might want to think twice before starting to learn the C language. However, assuming that you can get sample source code in equal amounts for all available IIGS programming languages, which one should you pick? This is a good question. But, before we can answer it, we need to discuss the issue of the *environment* that you will be programming in.

Basically, there are two programming environments in the IIGS world: The ORCA environment, and everything else. "Everything else" has, at one time or another, included such programming products as TML Pascal (I and II), the Software Development Environment, GS-Forth, and, on the Macintosh, MPW with the IIGS cross development tools. However, these days, most of those products have disappeared, leaving "everything else" to be represented by the Merlin assembler.

Merlin is a venerable Apple IIGS product that has the advantage of being the fastest danged assembler ever created for the Apple IIGS. However, Merlin is an island unto itself, with little or no way to mix its files with those created by other language products. That's one of the main reasons we don't use Merlin here at *GS+* Magazine, and it's one of the main reasons I won't be discussing it in this series of articles.

The ORCA environment (which includes compatible environments like APW and GNO), on the other hand, was built from the ground up to be extensible; allowing you to not only add new languages to your programming environment, but new utilities and functionality as well. You don't like the editor that is built into ORCA? Install a public domain one that someone else has written or write your own! Want to write part of your program in ORCA/Pascal and another part in ORCA/C? Go ahead! As long as you are working with ORCA compatible languages, you can mix and match to your heart's content.

Now, let's look at each ORCA language, and see which one is best suited for the beginner:

• ORCA/C: If you already have some programming experience, and you have always wanted to learn C (after all, it looks great on a résumé), you could probably start your IIGS programming career with C. One of the nice things about C is that there are a multitude of general instruction books for it, and literally millions of lines of sample source code out there for you to peruse. The Byte Works also offers a beginner's programming course built around

ORCA/C, Learn to Program in C. However, if you are an absolute rank beginner, C would probably be better as a second or third language. C is a very flexible and powerful language, but, it expects you to know *exactly* what you are doing, and it doesn't forgive mistakes.

• ORCA/M: ORCA/M isn't a compiler for the language "M" (there is no such language), it's an assembler for 65816 assembly language code. If you don't have any idea what that last sentence meant, it's a pretty good indication that you should *not* start out with ORCA/M. Frankly, ORCA/M is best suited for very experienced programmers.

• ORCA/Modula-2: Modula-2 is a language that is a descendant of Pascal. So, they have a lot in common. However ORCA/Modula-2 is a relatively new language for the IIGS, so, at this point, there is very little sample source code written in it. Because of that, and the fact that there is no Modula-2 specific programming tutorial from the Byte Works, I can't recommend Modula-2 as a beginning language for the IIGS programmer.

• ORCA/Pascal: Well, this is the last ORCA language that there is, so as you might expect, this is the one I'm going to recommend for beginners. There are several reasons for this: Pascal is a very popular language with schools, so there are lots of instructional books available for it, there is a lot of Pascal source code out there for you to peruse, and the Byte Works makes a Pascal programming tutorial, Learn to Program in Pascal, intended specifically for use with ORCA/Pascal. So, if you are a true beginner, you really can't go wrong with ORCA/Pascal.

Yet More Books

After you pick a language, you will need a good reference book (or books) for it. Without a good general language reference, you will be at the mercy of the ORCA manuals, and unfortunately, they don't tell you that much about the languages themselves. So, a good general language reference is a must. I've listed some of the ones that I'm familiar with at the end of this article, but don't be afraid to ask the opinions of your programmer friends online or at your user group.

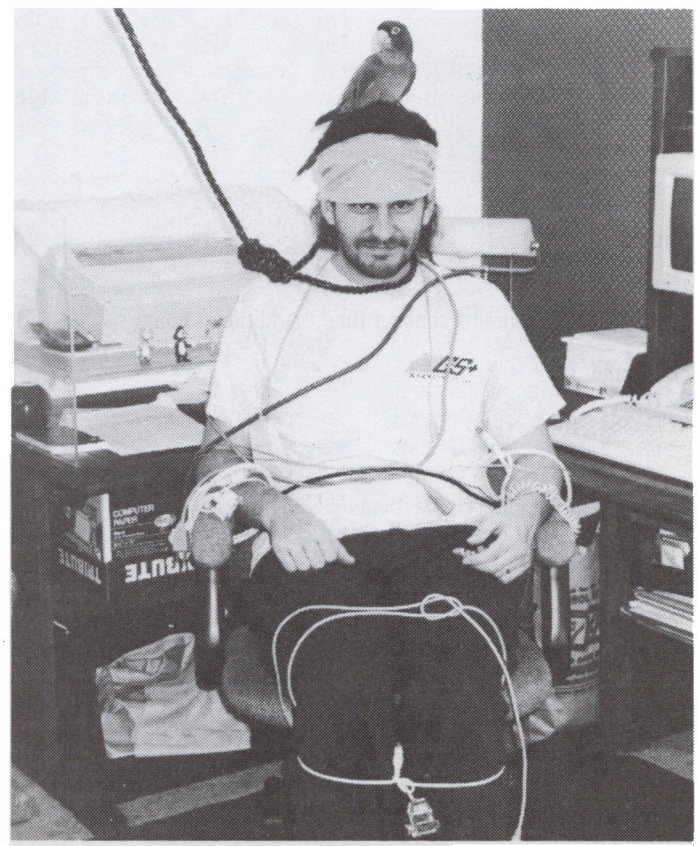
Here it is Again, in Case You Missed it.

So, now that I've gone over all the things that you need, let's put them all together into a step-by-step description of how to get started programming the IIGS.

1. Buy more memory and a hard disk. Make sure you have at least 4MB of RAM and at least a 40MB hard disk.
2. Make sure you have enough programming experience. If you don't, get ORCA/Pascal and the Learn to Program in Pascal tutorial, and go through the tutorial to the end. If you already have lots of experience, don't forget step 6, otherwise, stick with ORCA/Pascal and just skip step 6.
3. Get all the books you need.
4. Join an online service and/or user group. Use these memberships to get your hands on as much sample source code as you possibly can.
5. Get as much supplemental material as you can. Back issues of *Nibble*, *8/16-Central* and *GS+* Magazine are a good place to start.
6. Pick a language.
7. After you pick your language, read the manual and go through all of the sample source code that comes with it.
8. Buy a good general reference manual for the language you picked.
9. Decide on a *small* program that you want to write, and get started on it. When you finish, think of a slightly bigger, more complex program and then get started on it.
10. Keep programming. Remember, programming is a skill, if you don't use it, you lose it.

That, in a Nutshell, is it!

As you might be able to tell from the proceeding items, programming the IIGS is no easy task when you first get started. It takes time, money, dedication, and commitment. However, I'm not pointing all of this out to discourage you. I just want to make sure you know that, if you want to program your IIGS, it isn't something you can enter into lightly. You *will* spend a lot of money on books and compilers. You *will* sometimes feel that you just *can't* figure something out. You *will* spend a lot of time banging your head against a wall trying to figure out what's wrong with your programs. But, you *will* have a lot of fun doing all of these things, and, if you stick with it, you will experience a great feeling of satisfaction when you finish your first program, and your second, and your third .... If you can stick with it, programming your IIGS to do what *you* want it to do could just be the most exciting thing you ever do with your computer. ***GS+***



Okay guys, the photo shoot is over. Somebody get me out of this. Hey, what do you mean you're all going out to lunch?

Contact Information
(Prices Shown Are Retail or Retail/Mail-Order When Applicable)

Books

Apple IIGS Toolbox Reference: Volume 1 - $26.95
Apple IIGS Toolbox Reference: Volume 2 - $26.95
Apple IIGS Toolbox Reference: Volume 3 - $39.95
Apple IIGS GS/OS Reference - $28.95
Programmer's Reference For System 6.0.1 (includes the *Programmer's Reference For System 6*) - $49.95

Available from:
Resource Central
P. O. Box 11250
Overland Park, KS 66207
913-469-6502

Macintosh Human Interface Guidelines - $29.95

Available at your local bookstore, or contact:
APDA
Apple Computer Inc.
P. O. Box 319
Buffalo, NY 14207-0319
800-282-2732

C Primer Plus
By Mitchell Wait, Stephen Prata, Donald Martin

Oh My! Modula-2!
By Doug Cooper

Oh! Pascal!, second edition
By Doug Cooper and Michael Clancy

Online Services

America Online
$9.95/month, includes five hours of connect time. Additional hours are $3.95 each.
8619 Westwood Center Dr.
Vienna, VA 22182-2285
Voice: 800-827-6364

Delphi
Delphi offers several different pricing plans. Call for the latest pricing information.
General Videotex Corporation
Three Blackstone St.
Cambridge, MA 02139
Voice: 800-544-4005

GEnie
$8.95/month, includes four hours of non-prime time usage. Additional non-prime time hours are $3.00 each.
401 N. Washington St.
P. O. Box 6403
Rockville, MD 20849-6403
Voice: 800-638-9636

Programming Languages & Tutorials

ORCA/C - $150/$80
ORCA/Pascal - $150/$80
ORCA/M - $125/$70
ORCA/Modula-2 - $150/$80
Learn to Program in C - $50
With ORCA/C - $200/$120
Learn to Program in Pascal - $50
With ORCA/Pascal - $200/$120

Available from:
The Byte Works, Inc.
4700 Irving Blvd. NW, Suite 207
Albuquerque, NM 87114
505-898-8183

Also available from:
Resource Central (see above), and,

Quality Computers
20200 Nine Mile Rd.
St. Clair Shores, MI 48080
800-777-3642

Professor Gumby's Top Ten Rumors of All Time
Number 5 - From *GS+* V2.N6
July-August 1991

The Merchandising Continues . . .

Due to the runaway success of the new ***GS+*** T-shirts (get yours today!), the publisher has decided to expand the items available in the ***GS+*** Magazine merchandise catalog. In the next few months, you'll be seeing ads for ***GS+*** Mugs, ***GS+*** Pencils, ***GS+*** Matchbooks, and a complete line of fully posable, ***GS+*** "Editorial Action Figures." That's right, now you can pit your very own Nory™ and Susi™ action figures against your very own "Gonzo"-Joe™, Big Dave™ and Diz™ action figures. Can Nory and Susi convince these slothful Neanderthals to complete their articles on time? The fate of ***GS+*** Magazine is in *your* hands! ("Kung Fu Grip" figures will be available at an extra cost.)

Apple II Peripherals - Upgrades - Service Parts - Hard Disks
School & University P.O.s Accepted! - School Qty. Discounts Available*

RAM Chips

RAM Chips for your....
Expanding your... Apple IIgs Memory Card, Æ GS-RAM, Æ RAMWorks, Super Expander & Compatibles
768K Exp. Set (24 41256) $24.00
41256.....$1.25
Expanding Your GS Juice+, S&S ÆGS RAM + / II , CV Tech or Chinook 4000
41024 (1 Meg) $3.00
Bank of 8 .....$24.00
Expanding Your Æ GS-RAM Ultra or RAMPak4 GS
44256 (256x4) $3.00
Bank of 8 .....$24.00
Expanding Your OctoRAM, GS Sauce, Q-RAM GS, Pacemark 4 Meg RAM Card
1 Meg SIMM $24.00
RAMFast / CV Tech RAM 1Meg
Upgrade Set 2 441024-80.....$40.00

PC Transporter

101 Keyboard for PCT $29.00
PCT IBM Keyboard Cable.....$7.00
MS DOS 5.0 Package.....$49.00
3.5" 720K Ad-on Drive.....$49.00
4464 Zip /PCT Mem. Exp. ....$2. ea.

Motherboards/Service Parts

IIgs Power Supply Exch $59.00
IIgs Power Supply New $79.00
IIgs Keyboard Exchange $45.00
IIgs Keyboard Refurb $55.00
IIgs Keyboard NEW.....$69.00
Apple //e or //c Case.....$10.00
IIgs Case $25.00
ROM 01 Motherboard Exch..$149.00
ROM 03 Motherboard Exch..$199.00
Upgrade: ROM 01 to ROM 03
W/Working ROM 01.....$125.00
W/NonWorking ROM 01 .....$175.00

Refurbished Printers

C-Itoh ProWriter Jr (9 Pin) $84.00
ImageWriter I.....$129.00
ImageWriter I (Wide) $139.00
ImageWriter II $199.00
9 Pin Parallel Printer.....$75.00
IW-I/II Ribbons.....6 for $12.00
Print Heads
IW-I (Exch Only) $35.00
IW-II (Exch. Only).....$79.00
Epson 9 Pin Heads (Most) $59.00
We rebuilt print heads

Accessories

IIgs Internal Fan.....$14.95
9 Pin Joystick , IIgs.....$12.95
Mouse Pad (Static Free) $2.95

Cables & Switch Boxes

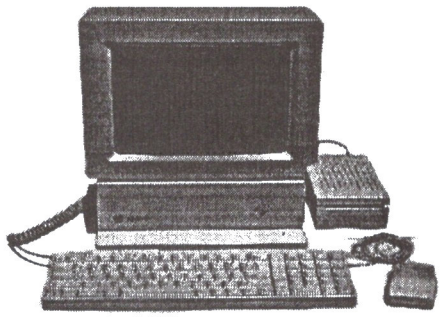
METAL, GBBS Sysops. HST/v.32bis hardware hndshk cables! (Specify)....$14.95
19 Pin Drive Converter.....$14.00
IIgs,e,c to ImageWriter I/II...$7.95
Fullnet Connector $14.95
Switch Boxes..... $19.00 - $29.00
ADB Replacement cable.....$9.95
SCSI Cable, 25-50 or 50-50.....$9.95

Peripheral Cards

Zip GS 7Mhz / 32K Cache ..$159.00
Zip Chip 8 Mhz (//e, //c) $139.00
AEC SCSI Card.....$59.00

Manuals & Software

Misc. Software. Games, Utils, More. $5.95 to $19.95. Ask For List.
Apple Manuals Avail. Ask For List



System Includes:
AppleColor RGB Monitor
External Apple 3.5" Drive • Keyboard
Mouse • 1024K
System 6.0.1
ROM 03

Apple IIGS System.....$599.00

Cordless ADB Mouse 6 foot range. Works Great! $59.00
Apple ADB Mouse II (New) $69.00..Refrb ADB Mouse .$49.00
Apple IIgs System 6.0.1 Disk/Manual Set $35.00
Apple IIgs System Transport Case (Full System).....$19.95

Vision Plus Enhanced IIGS $209.00 IIGS

Includes Allison™ Digitizing Software. Real Time Video Digitizer. 30 FPS B&W. Movie Mode & More! Upgrades available for current Vision Plus Owners. Call for details. Allison Upgrade...$49.95
Actual Screen Image

Modems

USR 1200 Modem (NEW).....$19.00
Hayes SM1200 (Refurb).....$29.00
2400 Ext & Cable MNP5 .....$69.00
Omron 14.4 Ext MNP1-7.....$149.00
Cable/Modem Combo.. add $5.00
Modems include ProTerm 3 Demo.

Bulk Disks

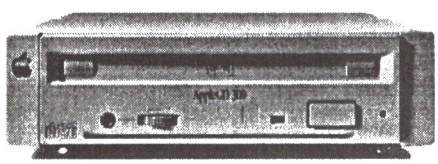
3.5" DSDD Disks 50 for $14.50
3.5" DSHD Disks 10 for $3.90

Alltech Electronics Co.

602 Garrison St. Oceanside, CA 92054
9-6 PM Mon.-Fri. - 9-5 PM Sat.

Orders Only: VISA MasterCard
800/995-7773

All other Inquiries: & Tech Support: 619/721-7733
Fax (School PO's, Quotes, Info.): 619/721-2823
BBS (FutureNet #30) 300-14.4 v.32: 619/721-0705



Great for DiscQuest!

Apple CD 150 $199.00

Monitors

IIgs Compatible (Atari) RGB Monitor.....$129.00
AppleColor RGB, IIgs (Refurb).....$169.00
AppleColor RGB IIgs (NEW) $225.00
Apple Color Comp or Color 100 (exch) ...$109.00
Composite Green (Refurb).....$35.00
Composite Green "Box" Monitor 9".....$39.00
Apple Monitor][or /// (Refurb).....$49.00

SEQUENTIAL SYSTEMS

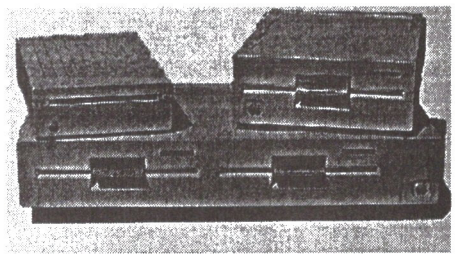
Sequential Systems RAM-GS 4096K .....$119.
Meg 80 Z (1 Meg, Apple //e).....$59.00
Q:Talk LTO (AppleTalk & 32K for IW-II) ..$69.00
RAM //c with 1024K (//c or //c+).....$119.
bit Mouse card $49.00 ....with mouse.....$65.00
SuperCOMM $54.00.....SuperCOMM /c $45.00

RAM Cards

Apple 1 Meg Memory Exp. with 1 Meg.....$39.00
GS SuperRAM 4 Meg with 1 Meg.....$74.
GS Super RAM w/2 Meg $99.00. 4 Meg$129.
CV Tech Memory Board 4 Meg $205. 5-8Call

SCSI Hard Disks

40 Meg External w/AEC Card (Refurb).$159.
40 Meg External Drive (Quantum).....$139.
80 Meg Ext.....$189.00 w/Apl SCSI.....$295.
SyQuest 44.....$259.00 SyQuest 88. .....$359.
External Drive Cases
SCSI Case & P/S for 3.5" or 5.25" HH.....$49.
SCSI Case & P/S for Full Hight 5.25".....$69.



Apple FDHD $199.00

Disk Drives

5.25" Generic Full Height w/19 Pin $69.00
5.25" Apple Disk II (Refurb) add $10.00 for above.
Apple DuoDisk 5.25" Exch.....$129.00
Apple DuoDisk 5.25" (Refurb) $159.00
AppleDisk 3.5" (Platinum).....$165.00
AppleDisk 5.25" (Refurb).....$165.00
Apple 3.5" 800K Drive Service Exchange $95.00

CA Residents please add 7.75% Sales Tax • COD Orders, Cash/Certified Funds Only • Minimum Shipping & Handling Charge $5.00 (This covers most orders) • Some products are refurbished products and carry a 120 day warranty. • New products are covered by a one year warranty. All Repairs are warranted for 120 days. • Some RAM Chips may be new but pulled, all RAM has a one year warranty. • NO Returns without an RMA number, please call before shipping anything back to us. • A 20% Restocking Fee is applied to all returns due to incompatibility or unneeded product or Products purchased in error. Please be sure what computer you have before placing an order. • Educational P.O.s Accepted, Others on AOC only. • Prices and availability subject to change at any time without any notification whatsoever. *Discounts on certain merchandise only and min. Qty Applies.

GS+ Back Issue Information

May-Jun 1990 (V1.N5)
- AppleFest Report
- Beginner's Guide to System Disks - Part 1
- GS/OS prefixes - PreFixer CDev
- Brush with Greatness - How your IIGS makes colors
- Reviews: CMS 45MB Removable Hard Drive, S&S-RAMCard, DataLink Express modem, Visionary GS digitizer, GraphicWriter III, ZapLink, McGee, Math Blaster Plus IIGS, The New Talking Stickybear Alphabet, ZipGS

Jan-Feb 1991 (V2.N3)
- AppleFest/Long Beach '90 & Apple II Achievement Awards
- Interview with Jim Carson of Vitesse, Inc.
- Introduction to System Software v5.0.4
- RAM Namer - A CDev that allows you to rename RAM disks
- *GS+* program updates: Battery Brain v1.1, EGOed v1.32c, Teach Translator for GraphicWriter III v1.1
- Reviews: ZipGSX, LightningScan, Design Your Own Home, Print Shop Companion IIGS, Your IIGS Guide, Dragon Wars, 2088: The Cryllan Mission - Second Scenario, Space Ace, Sinbad & the Throne of the Falcon

Sep-Oct 1991 (V3.N1)
- Protecting Your Investment - A Guide to Surge Protection
- A Conversation with Roger Wagner - Part 2
- Working with the Toolbox - Part 4: QuickDraw II
- FGS - A desktop program that generates Fractals
- *GS+* program updates: EGOed v1.36, Autopilot v1.1, NoDOS v1.6
- Reviews: two 100MB hard drives, Nite Owl Slide-On Battery, ORCA/Integer BASIC, ORCA Talking Tools, Storybook Weaver: World of Adventure HyperBole, HoverBlade, Shareware: DeskTop Painter, SoundSmith, IIGS Classic: Bard's Tale IIGS

Jul-Aug 1992 (V3.N6)
- KansasFest 1992
- Introduction to 3-D Graphics - Part 3: Speeding Things Up
- Working with the Toolbox - Part 8: The Control Manager
- Understanding FSTs
- Using rBundles in Your Programs
- Quick Folder - A Finder Extension that allows you to open folders from the Finder's Extras menu. **Requires System 6.**
- Extra Bits - A Control Panel that lets you change the new Battery RAM parameters that System 6 didn't provide a Control Panel for. **Requires System 6.**
- *GS+* program updates: EGOed v1.7 **(requires System 6)**, Quick DA v2.0 **(requires System 6)**, Replicator v1.3
- Reviews: ZipGS (10MHz CPU/64K Cache), Gate, Space Fox, Utility Launch & Utility Works

Sep-Oct 1992 (V4.N1)
- Apple EXPO East
- Open From Desktop - A Finder Extension that allows you to open any item on your desktop from the Finder's Extras menu. **Requires System 6.**
- II Notes - A 20-page NDA notepad. **Requires System 6.**
- Miscellaneous Library - A collection of useful routines to use from any programming language that supports linking to standard libraries
- *GS+* program updates **(require System 6)**: Autopilot v2.0 , Quick DA v2.1, EGOed v1.7.1
- Reviews: ContactsGS, GSymbolix, Kangaroo, ORCA/Debugger, UltraCat, Storybook Weaver: World of Make-Believe

Nov-Dec 1992 (V4.N2)
- Understanding Accelerators
- The Basic IIGS
- Working with the Toolbox - Part 9: The Menu Manager
- Font Reporter - A program that lets you display and print out any font in your system. **Requires System 6.**
- Miscellaneous Library (updated)
- *GS+* program updates: EGOed v1.8 **(requires System 6)**, Replicator v1.3.1
- Reviews: AutoArk, 1990 GEM Apple II CD-ROM, IIGS System Transport Case, Out of This World, TrueType Font Collection, Universe Master
- Review updates: Desktop Enhancer v2.0, Pointless =v2.0

(All programs after this issue require System 6, unless otherwise noted)

Jan-Feb 1993 (V4.N3)
- The World at Your Fingertips
- Understanding the Desktop
- Batt Reporter - A program that generates plain English reports from battery RAM configuration files
- Rainbow - A Finder extension that lets you change the colors of your device icons
- Miscellaneous Library (updated)
- *GS+* program updates: Battery Brain v2.0, Open From Desktop v1.0.1, Rebuild Desktop v1.1, EGOed v1.9
- Reviews: CV-Ram Memory Card, StyleWriter printer, ProSel-16, TransProg III v1.1, Ant Wars, FloorTiles,Quest for the Hoard

Mar-Apr 1993 (V4.N4)
- Beginner's Guide to Finder v6.0
- Working with the Toolbox - Part 10: LineEdit
- LASERbeam - A program that lets you download PostScript files to a PostScript printer
- Font Memories - A control panel that lets you keep your bit-mapped fonts on a disk other than your startup disk
- EGOed lite - a smaller, faster version of the EGOed New Desk Accessory
- Miscellaneous Library (updated)
- *GS+* program updates Rainbow v1.0.1, NoDOS v1.8
- Reviews: Salvation—Deliverance, DreamGraphix, The Manager, The Passport House Letter, The Lost Tribe, DuelTris

May-Jun 1993 (V4.N5)
- The Scavenger - Using your IIGS with CD-ROMs from other computers
- Apple EXPO West Report
- Anna Matrix - a Cool Cursor Editor
- *GS+* program update: Cool Cursor v2.0, Miscellaneous Library
- Reviews: Apple Desktop Bus Mouse II, Baccarat, Key Fonts Pro CD-ROM, MAZER II: The Ghost of Mordaine, Pick 'n' Pile, Shanghai II: Dragon's Eye, Solarian GS, Twilight II, TypeWest Volume 1

Jul-Aug 1993 (V4.N6)
- System 6.0.1—For Users
- KansasFest 1993
- Catch the .WAV: A Guide to Scavenging Sound Files
- Secrets of Writing Twilight II Screen Blankers
- Finder Binder: Avoid the annoying "An application can't be found for this document" dialog by connecting documents to an application
- *GS+* program updates: AutoSave v2.0, EGOed lite v1.0.1, Extra Bits v1.0.1
- Reviews: Castle Metacus, HardPressed, The Lost Treasures of Infocom, Treasures From Heaven: Quest for the Hoard 2, Your Money Matters, Zip Drive

Sep-Oct 1993 (V5.N1)
- So You Bought a Hard Disk . . . Now What?
- Apple (Jive) Talkin'
- An Introduction to Object Oriented Programming
- File Dump: A complete Object Oriented Programming example written in ORCA/Pascal v2.0.1.
- *GS+* program updates: Anna Matrix v1.0.1, Cool Cursor v2.0.1
- Reviews: Applied Engineering's High Density Disk Drive, Apple II SuperDrive Controller Card, MODZap, soniqTracker, ORCA/Pascal v2.0.1, SoundMeister, TypeSet

Nov-Dec 1993 (V5.N2)
- IIGS Maintenance–Part 1: The Mouse and Keyboard
- SCSI ("Simple Connections," Says Igor.)
- Balloon v 1.0: A finder extension that lets you extract files from ShrinkIt Archives
- CD-ROaM: An application that lets you scavenge files off of CD- ROMs
- KaBlooie! A version of the classic game Minesweeper for your IIGS
- Reviews: 3D Logo, Focus Drive Hard Card, Prism, Tulin Floptical Disk Drive

Jan-Feb 1994 (V5.N3)
- IPC (Igor's Playful Code) - A guide to using IPC on the IIGS
- EGOed v2.0: Read and write RTF files, plus a new color menu
- MIDI Surgeon: Convert MIDI data files to MIDI Synth format
- Reviews: Ancient Glory, Apple Extended Keyboard, AudioClips, GNO/ME 2.0, HP DeskWriter 550C Printer, HyperLogo, NCS Pro 240 Hard Disk, Pedigree

Any issues that are not listed are sold out.

Prices
$5 For Each Magazine
$6.50 For Each Disk
$10 For Each Magazine and Disk Set

All prices include postage and handling. (Orders will be sent First-Class to the U.S., Air Mail to Canada and Mexico, and surface mail to all other countries. For Air Mail to all other countries, add an additional $3 per issue. If you pay by credit card, you will be charged only for the additional postage needed.) *Tennessee residents must add 7.75% sales tax.* Mail back issue requests to: *GS+* Back Issues, c/o EGO Systems, PO Box 15366, Chattanooga, TN, 37415-0366; or call 1-800-662-3634 (or 615-843-3988), Monday through Friday, 9 a.m. to 5 p.m. Eastern Standard Time, to order or verify availability. Please include your phone number on all orders placed by mail (in case we are sold out of an issue)! For MasterCard or VISA orders placed by mail, also include your credit card number, expiration date, and signature.

Playful v1.0

By Josef W. Wankerl

In our last issue, I presented a nice sample program to demonstrate how Inter-Process Communication (IPC) works [in the article, "IPC (Igor's Playful Code)" - Ed.]. The program was a small Finder extension which would play a sound whenever you opened a file. As always, Diz looked at the program and wanted more. "You've got all the routines there to make it so that when you double-click on a file in the Finder, all the `rSoundSample` resources contained in the file would be played," said Diz. "Well I guess so," I replied, knowing that the only routine in there which would be used was the one that contained the request to play the sound, plus a little common Finder extension code. (The boss is always right, you know.) So the program was assigned to me for this issue. Although it wasn't difficult to write, it was a lot more work than simply tacking on a couple of lines in the IPC sample program. But hey, enough of my yakkin', let's rock-n-roll.

Installing Playful

To install the Playful Finder extension, refer to "How to Use Your *GS+* Disk" elsewhere in this issue. Playful can *only* be used with System 6.0.1—it will *not* work with a system prior to 6.0.1. (If you don't yet have System 6.0.1, see "How to Get System 6.0.1" elsewhere in this issue.)

Playing rSoundResources

Once you have Playful installed, all you need to do to play all the sounds contained in a sound resource file is to simply open the file in the Finder. You can probably find a lot of sound resource files in your ***:System:Sounds** folder, but they don't have to be there for Playful to recognize them. In other words, you no longer have to use the Sound control panel to listen to all your sounds!

As it is playing the sounds, Playful will display the name of the file the sound is contained in as well as the name of the sound it is currently playing.

Sounds Everywhere

Did you know that sound resource files aren't the only place that you can find `rSoundSample` resources? In fact, just about any file could have sounds hiding in it! For example, some games (like our Kablooie! game from *GS+* V5.N2) keep the sounds that they use inside their own resource forks. So, what if you want to hear those sounds, but you don't want to play the game? With Playful installed, it's no problem! Playful adds a menu item, "Play rSoundSamples," to the Finder's Extras menu. You just select the files (again, it can be *any* type of file!) that you suspect have sounds in them, and choose the "Play rSoundSamples" menu item. Playful will load all of the `rSoundSample` resources contained in the selected files and play them for you!

That Other Program

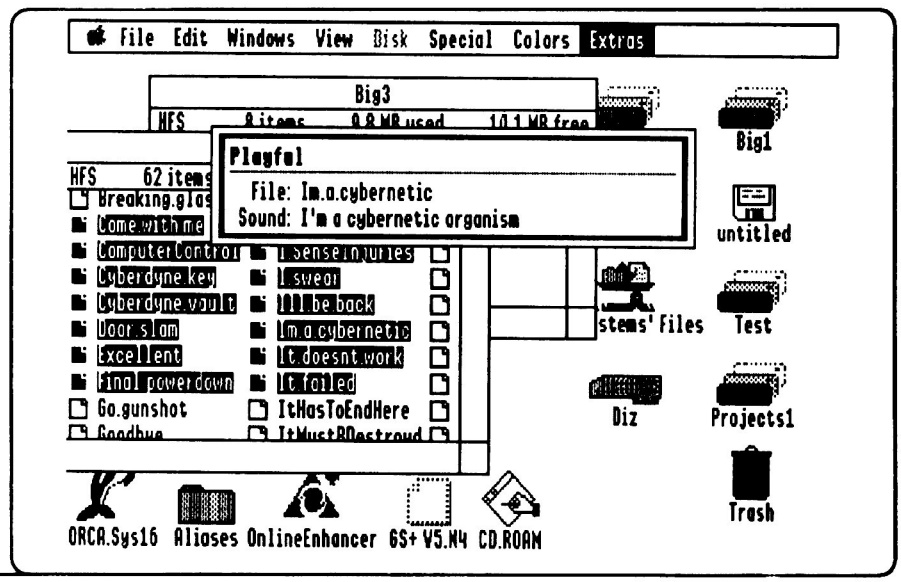
Some of you may already know of a freeware program called Finder Sounder which does a similar job of playing the sounds in sound resource files. There are a couple of things that Playful does that Finder Sounder doesn't: Playful tells you the name of the file that sounds are currently being played from, it tells you the name of the sound that is playing, it will play the entire sound before proceeding to the next sound, it will play *all* of the sounds in the file, and it will play `rSoundSample` sound resources from inside files that are not explicitly sound resource files.

Programming Tricks

One of Playful's neater tricks is its consistent handling of both opening files and the "Play rSoundSamples" menu item. What this means is that the window containing the current sound information is displayed until all of the sound files have been processed. For the menu item, this is no problem because Playful is in control all the time—all it has to do is get the selected icons, open the status window, process all the files, then close the status window. The tricky part comes when a bunch of files are opened. For each file, the Finder sends a `finderSaysBeforeOpen` request. Playful can then look at the file, process it, and then *return to the Finder*. This means it has to clean up after itself because it doesn't know if any more files are ready to be sent. Playful uses a simple trick to know when the Finder is done sending the `finderSaysBeforeOpen` requests: it watches for the `finderSaysIdle` request. The first time Playful sees a `finderSaysBeforeOpen` request, it opens the status window, processes the file, then returns to the Finder. If another `finderSaysBeforeOpen` request is sent, Playful knows the status window is open (it uses a global variable to keep track of the window) and doesn't re-open it if it is already open. When all the `finderSaysBeforeOpen` requests have been sent, the Finder will return to its main event loop. One of the jobs of the Finder's main event loop is to send out the `finderSaysIdle` request. When Playful sees the `finderSaysIdle` request, it checks to see if the status window is open, and if so, it closes it. Nifty trick, eh?

What Else?

That's basically all there is to Playful. (Unless you can find the "Easter Egg" that I put in it!) If you find a problem with Playful, please fill out the problem report form on your *GS+* Disk and send it in. *GS+*



APPLE II EXPANSION HEADQUARTERS!

WE HAVE THE PRODUCTS YOU NEED WITH THE PRICES YOU WANT!

HARD DRIVES

Our NCS Pro SCSI Hard Disk drives have all of these great features:

•2 year warranty (Quantum Mechanisms)
•High impact ABS case (not flimsy plastic material)
•Dual SCSI connectors for easy daisy-chaining
•Access indicator light
•Termination accessible via access door
•External SCSI ID selector

We also include the latest System Software for your Apple IIgs. You also get 30+ megs. of shareware and utilities at your disposal. Works great for other Apples too.

And if you're still not satisfied, just return the drive by taking advantage of our no-hassle return policy.

NCS Pro Drives

42mb.	Quantum ELS	$159
105mb.	Quantum SPECIAL	$199
170mb.	Quantum ELS	$239
340mb.	Quantum *NEW* LPS	$349
540mb.	Quantum *NEW* LPS	$579
1080mb.	Quantum *NEW* LPS	$949
1800mb.	Quantum *NEW* LPS	$1159

NCS Pro Removable Drives

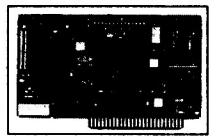
***Special* NCS Pro R44mb. $249**
44mb. Syquest Removable Drive

NCS Pro R105mb. $499
105mb. 3.5" SyQuest Read and Write

All removable drives include one cartridge.
Extra 44 / 105 carts are $68 / $89

HARD DRIVE INTERFACES

SEQUENTIAL SYSTEMS
RamFast SCSI Card



The RamFast is the most powerful SCSI interface available for the Apple IIgs Computer. It features expandable Cache, and upgradable ROM support.

RamFast GS v3.01
w/256k Rev.D **$139 / $149**
w/1mb.Rev.D **$189 / $199**
(with hard drive / without hard drive)

RamFast v3.01 Upgrade (for current RF users) Rev. C or D $15

TRADE-IN YOUR APPLE SCSI CARD FOR A RAMFAST SCSI CARD, CALL FOR DETAILS!

APPLE
High Speed SCSI Card $99
$109 without hard drive purchase

SEIKO DATA WATCH SPECIAL!

Put the scheduling power of your Apple II on your wrist. Allows you to keep daily details, appointments all on your wrist. Uses the easy to use Time-Trax 23-month scheduling software. For any Apple II computer! Includes interface cable, transmitter, Silver slim-line LCD watch, software and manual.

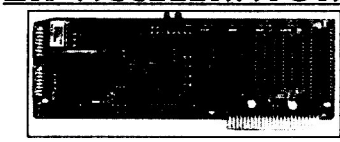
$24.95
DELIVERED OVERNIGHT! (MOST AREAS)
$19 WITH ANY OTHER PURCHASE

SEQUENTIAL SYSTEMS

RAM GS
(4megs/GS) $118*
*$129 without any other purchase
•4MB. ON BOARD FOR IIGS
•100% DMA COMPATIBILITY
•INCLUDES DIAGNOSTIC UTILITIES DISK

PaceMark IIgs SuperRam IV $99*
4mb. ram card for IIgs - 100% compatible!
*109 without any other purchase

ZIP ACCELERATORS



*Put your GS into the fastlane with ZIP Accelerators. Over **300%** faster with a Zip 9/32 installed. Makes your GS Finder like new!*

Zip 7mhz/8k Cache	**$139**
Zip 8mhz/16k Cache	**$174**
Zip 9mhz/32k Cache	**$235**
Zip 10mhz/64k Cache	**$379**

LRO Computer Sales

DISCOVER NOVUS | VISA | MasterCard | Checks

School and Government P.Os Welcome!

Serving the Apple II Community for over 5 Years!

VITESSE

Quickie / Inwords Bundle.....$199

Includes Vitesse's Quickie Handscanner for the Apple II and WestCode Software InWords v1.1 OCR Software.

Quickie (separately).....$149

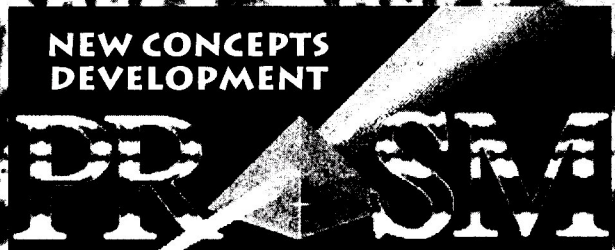
HEAVY DUTY POWER SUPPLIES

Apple IIgs H-D Power Supply $79

Apple IIe H-D Power Supply $69

All Heavy Duty Power Supplies have a 1-year Warranty!

NEW CONCEPTS DEVELOPMENT



IIgs 3200 Color Graphics Multi-Conversion Utility **$34.95**

Applied Engineering

- **ÆVulcan Gold Upgrade.....$ 47**
- **Æ 5.25" for LC or A2.....$149**
- **AMR 5.25" for LC or A2....$119**
  -AMR and Æ 5.25" drives are daisy chainable !-
- **Æ SuperDrive.....$229**
  -1.44/800K 3.5" Floppy Drive for A2 & Mac
- **Apple SuperDrive I/F.....$138**
  -3.5" interface for A2 and IIgs / Req'd for 1.44mb.

Memory Upgrades

- **1MB SIMMs.....$46**
- **1MB RamFast Upgrade.....$59**
- **1MB Expansion Set (8 Chips).....$49**
- **1MB GS-Ram III Upgrade.....$59**
- **256k GS-RAM Ultra Upgrade.....$15**
- **256k Apple Exp. Kit.....$18**

BRAND NEW! SEQUENTIAL SYSTEMS DiscQUEST™

SEQUENTIAL SYSTEMS DiscQUEST™ Software bundled with a CD-ROM drive allows **YOU** the Apple IIgs user to explore the hundreds of CD-ROM disks previously unaccesable.

SEQUENTIAL SYSTEMS DiscQUEST™ has powerful features you would not expect, for instance:

- Search CD-ROMs by Word, Title, Author
- Display Graphics from Mac or IBM CD-ROM disks
- Play audio from CD-ROM disks
- Browse through Mac and IBM CD-ROM disks

SEQUENTIAL SYSTEMS DiscQUEST™ CD-ROM Bundle **$569**
-includes Software, RamFast SCSI, CD-ROM Drive and 4 CD-ROM titles!
SEQUENTIAL SYSTEMS *DiscQUEST™ Software alone* **$79**
-includes Family Doctor CD-ROM

SEQUENTIAL SYSTEMS

Switch-It!.....$48.00
Switch-It! allows you to run several applications at once, and switch between them in an instant. Compatible with practically all GS/OS programs. Requires GS/OS 6.0, and 4MB for optimum performance (2MB minimum)

DREAMWORLD

DreamGrafix.....$59.00
-3200 COLOR PAINT PROGRAM FOR IIGS

ECON TECHNOLOGIES

Universe Master.....$64.95
-COMPLETE IIGS HARD DRIVE MANAGEMENT UTILITY
AutoArc.....$24.00
-NEARLY DOUBLES THE SIZE OF YOUR IIGS HARD DRIVE

Roger Wagner Publishing

HyperStudio.....$97.00
-FUN AND EASY TO LEARN HYPERMEDIA FOR THE IIGS

WestCode SOFTWARE

Pointless.....$46.00
-TRUE TYPE FONT INTERPRETER FOR IIGS
InWords.....$78.00
-ALLOWS THE QUICKIE TO SCAN IN TEXT FOR EDITING

Vitesse

Harmonie.....$29.00
-IIGS PRINTER DRIVERS FOR HP PRINTERS

InTrec

ProTerm 3.1 NEW.....$78.00
-HIGHLY RATED APPLE II COMMUNICATIONS PROGRAM

TimeWorks

Publish It! v4.0 **SPECIAL.....$29.00**
-APPLE II Desktop Publishing Program

Seven Hills Software

(ALL COMPATIBLE WITH THE MANAGER)
Shoebox.....$34.00
-HOUSEHOLD RECORD KEEPING AND MESSAGING
Express.....$24.00
-IIGS PRINT SPOOLER
Kangaroo.....$28.00
-TIMESAVING FOLDER UTILITIES FOR IIGS
Manager.....$48.00
-MULTITASKING ON THE IIGS
SPECTRUM (NEW in stock)..$84.95
-SYSTEM 6.0 BASED COMMUNICATIONS PROGRAM, SUPPORTS ZMODEM AND COMPUSERVE B+ PROTOCOLS!

Apple Computer, Inc.

GS/OS System 6.0.....$39.00
HyperCard GS.....$69.00

// Productive

BullsEye TrueType Font Collection for the Apple IIgs. Includes over 300 TrueType Fonts, Font Viewing Utility, America Online Software Kit and MUCH MORE for only $45.99

// Productive BullsEye

and

WestCode Pointless $88.00

MODEMS

SUPRA v.32bis, 14.4bps Send / Receive Data/Fax Modem (Universal).....$209

Zoom 2400 with Cable for IIgs.....$79

Æ ReadyLink Communication Software only $39 with Modem Purchase

(800) ASK-4LRO

(2 7 5 - 4 5 7 6)

FAX ORDERS: (815)338-4332
TECH SUPPORT: (815)338-8746
INTERNATIONAL: (815)338-8658
ILLINOIS RESIDENTS: (815)338-8685
224 WEST JUDD STREET
WOODSTOCK, IL 60098

What Is This?

By Steven W. Disbrow

We Get Phone Calls

Actually, we get *lots* of phone calls. While some of these are questions about *GS+* Magazine or the programs that we publish, a large percentage of these calls are from people having trouble actually using their computer. Typically, these calls are "How do I use this file" calls. So, about a year ago (maybe longer—the days tend to run together when you are self-employed), I came up with the idea for a program that would provide information about any type of file that you might find on the IIGS.

So, What Is This?

What Is This? is a Finder extension that provides you with information on any icon that you can select in the Finder. All you have to do is select the icon that you are curious about, and then pick the What Is This? menu item from the Finder's Extras menu. You will then be presented with a window containing general information about the icon you have selected. If possible, What Is This? will not only tell you what the icon represents, but how to use it as well! You can also use What Is This? to view and change any comments associated with the icon.

Installing What Is This?

What Is This? requires System Software v6.0.1 or later. It will *not* work with System 6! To install What Is This? on your system disk (a hard disk is recommended), simply run the Installer program that is on your *GS+* Disk. For more information, and an example of the installation process, see "How to Use Your *GS+* Disk" elsewhere in this issue.

General Information

The general information presented by What Is This? varies depending on the type of icon that is selected. If the icon is a disk, device, folder or the Trash, What Is This? tells you what the icon is (in the case of a disk, it also tells you what file system the disk is formatted for, along with the file naming conventions for that file system), as well as telling you how to use the icon (i.e. how to throw things into the Trash can, how to open a folder to see the files inside it, etc.).

However, if the icon represents a file, the information presented is as detailed as I could possibly make it. First, the general type of the file is presented along with it's hexadecimal file type number. Then, if available, more information is presented explaining exactly what kind of information is in the file, along with such things as the application that would be needed to use the file, and how to contact the manufacturer of that application. (Note that this sort of detail is not available for every file. I could only include this much detail for those file types that I knew about, or could find out about through Apple's Apple II File Type Notes.)

Comments

On the IIGS, every file can have two comments about it in its resource fork. (And if a file does not have a resource fork, it can't have any comments associated with it.) The first type of comment is the one that you see when you use the Finder's Icon Info menu item on a file. This comment usually contains general information about the file, such as who created it, copyright information, etc. However, this comment *can* contain anything you want it to.

The second type of comment is one that you may have seen without realizing exactly what you were looking at: It is an informative comment that is displayed by the Finder when you try to open a file that the Finder can not find an application for. When this happens, the Finder looks inside the file for this secondary comment, and, if it exists, displays it. If the comment doesn't exist, the Finder puts up the annoying "An application can not be found for this document." window. This secondary comment should contain information on how to properly use the file. So, if the user tries to open the file, and is presented with this information, they should be able to follow the instructions in the comment and be able to use the file.

While this is an excellent concept, very few programs create files with this instructional comment present. So, What Is This? gives you a way to add this comment to files that don't have one; or you can edit this comment on files that already have one present.

So, those are the basics of what What Is This? can do. Now let's look at a screen shot, and discuss the controls in the What Is This? window.

The Information Box

At the top of the What Is This? window is a TextEdit control that contains the

Errata

Just in case you thought that last issue's cover made no sense whatsoever, you should know that you were correct. You see, under the photo, there was supposed to be a headline that read:

Sickly Sweet Valentine's Issue

So, I've included the above text so that you can photocopy this page, cut out the headline, and paste it on the cover of your magazine. This should make the cover much more understandable, and it's a lot cheaper than reprinting the issue and sending everyone a replacement copy.

Also, due to bad planning, this issue's continuation of the Top Ten Rumors of All Time includes only four rumors.

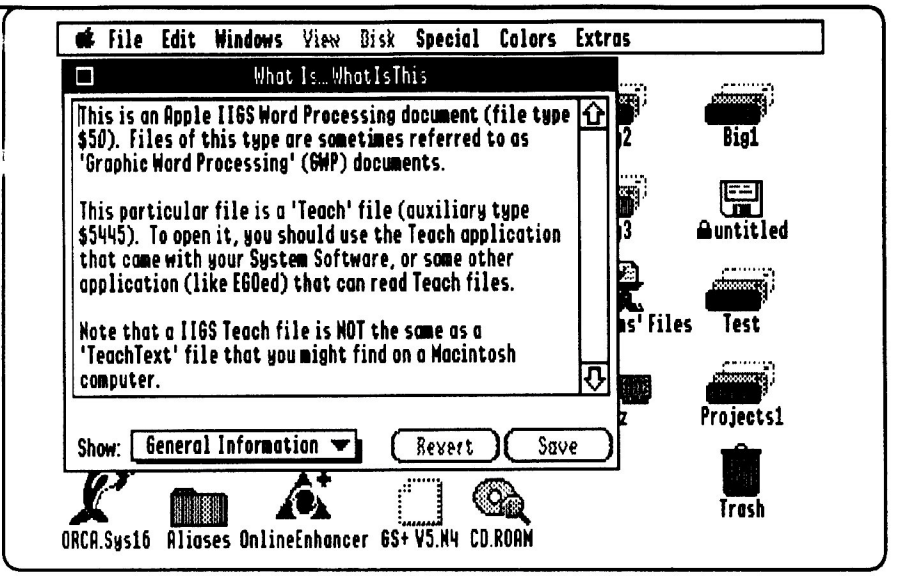
If you find a mistake in *GS+* Magazine, we want to fix it! Give us a call at (615) 843-3988 or write to us at:

GS+ Magazine
P. O. Box 15366
Chattanooga, TN 37415-0366

information that is available for the selected icon. When you're viewing the General Information for an icon, you will *not* be able to type anything into this information box. (Since the General Information applies to all icons of the same type, this information can't be changed for an individual icon.) Below the information box are a pop-up menu and two buttons. First let's look at the pop-up menu.

The Pop-Up Menu

This menu lets you choose which of the three types of information you wish to view for this icon. The default is "General Information." To choose a different type of information, simply select it from the pop-up menu. When you pick either the "Comments" or "Instructions For Use" item from this menu, What Is This? will load the appropriate comment from the file and display it in the Information Box. If the file does not have the comment you requested, What Is This? will tell you so, and give you instructions on how to add a comment to the file. What Is This? will also inform you if the file does not have a resource fork, and will warn you that adding a resource fork will render the file unusable by ProDOS 8 applications. And, in the case of a file that What Is This? knows to be a file used by a ProDOS 8 application, it will also inform you that adding a comment to the file is *not* a good idea!



The Revert & Save Buttons

When you are viewing a comment in the What Is This? window, you can change that comment to say whatever you wish. If, however, you decide that you don't want to make those changes, you can click on the Revert button and What Is This? will re-load the comment from disk.

When you *do* want to save your changes however, you must click on the Save button to make them permanent. If the file you are changing doesn't already have

Moving?

Well, don't forget to tell us! The Post Office does not normally forward Third-Class mail (they simply destroy it!), and we can't afford to replace magazines that were lost because a subscriber forgot to send us a change of address! If you miss an issue, we will extend your subscription, but you will have to buy the missed issue as a back issue! So, to avoid this hassle, send us a change of address as soon as you know your new address!

If you want, you can phone in your change of address by calling (615) 843-3988, or you can e-mail it to us at one of the following addresses:

America Online or Delphi: GSPlusDiz
GEnie: JWankerl
InterNet: GSPlusDiz@aol.com
(Be sure to include your old *address when e-mailing us a change of address!)*

Or, if you prefer to do things the old fashioned way, simply remove your mailing label from a previous issue of **GS+** *Magazine, affix it to a change of address form (available at your local Post Office), fill in your new address, and send it to us at:*

GS+ Subscription Services
P. O. Box 15366
Chattanooga, TN 37415-0366

a resource fork to hold the new comment, What Is This? will present you with a dialog to confirm that you really want to add the comment (and a resource fork) to the file. Remember, files with comments also have resource forks, and files with resource forks can't be used from ProDOS 8! (And, once a file has a resource fork, you can't easily get rid of it!)

That's About All There Is To It

When I was writing What Is This? my main focus was on providing as much useful information to the user as possible. Therefore, there aren't a lot of bells and whistles in it. However, a few little things that I threw in might make your life easier. For example, the standard editing keys for Cut, Copy, Paste and Select All work with the text in the Information Box. And, if you already have a What Is This? window open for a particular icon, and you select What Is This? for the icon again, the original window will be brought to the front, rather than opening up a second, duplicate window for that icon.

I Want Your Help!

I really think that you will like What Is This? and that, more importantly, you will find it genuinely useful and informative. However, as I said earlier, the information contained in What Is This? is limited by what I know from my own personal experience, and the currently available file type documentation from Apple Computer. So, if there is a file type that *you* know something about that I did not cover in this version of What Is This?, send that information in to me, and I'll try to put it in the next version! Be sure to include the file type, auxiliary file type, and contact information (if applicable), along with your description of the file. I'm especially interested in any information that you might have about the older applications that created some of these files. (By the way, if you think my descriptions stink and you have access to a resource editor, you can edit the descriptions in What Is This? to your hearts content.)

And, as always, if you have a problem with this program, be sure to fill out the *GS+* Problem form and send it in so that we can fix it as soon as possible. *GS+*

Warranty Disclaimer And Copyrights

EGO SYSTEMS' LICENSOR(S) MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING THE SOFTWARE. EGO SYSTEMS' LICENSOR(S) DOES NOT WARRANT, GUARANTEE OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE SOFTWARE IN TERMS OF ITS CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS OR OTHERWISE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE SOFTWARE IS ASSUMED BY YOU. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME JURISDICTIONS. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU.

IN NO EVENT WILL EGO SYSTEMS' LICENSOR(S), AND THEIR DIRECTORS, OFFICERS, EMPLOYEES OR AGENTS (COLLECTIVELY EGO SYSTEMS' LICENSOR) BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, AND THE LIKE) ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE EVEN IF EGO SYSTEMS' LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU. EGO Systems' Licensor's liability to you for actual damages from any cause whatsoever, and regardless of the form of the action (whether in contract, tort (including negligence), product liability or otherwise), will be limited to $50.

And *furthermore*,

Some of the programs on this *GS+* Disk contain materials from the ORCA/C, ORCA/Pascal, and ORCA/Modula-2 Run-Time Libraries, copyright 1987-1989 by Byte Works, Inc. Used with permission. All other material is copyright 1994 by EGO Systems, unless otherwise noted. Purchasers of the *GS+* Disk may use the source code and/or libraries in their own programs, provided that the following notice appears somewhere in the program or in its documentation:

"Portions of this program were created using source code and/or libraries from *GS+* Magazine, and are copyright 1989-1994 by EGO Systems. Used with permission. For more information on *GS+* Magazine, write to:
***GS+* Magazine**
P. O. Box 15366
Chattanooga, TN 37415-0366"

What's New

Compiled by Steven W. Disbrow

The MIDI Buying Guide

Now that you've played around with MIDI Surgeon (from *GS+* V5.N3), you may be wondering how you can make some MIDI files of your own. Well then, you should check out *The MIDI Buying Guide* from Sound Management publishing. What's in it? Well, let me just pull a quote directly from the press release:

"Included in this extensive MIDI Buying Guide are complete product descriptions packed with graphic-illustrations of MIDI software and hardware, educational MIDI-products, keyboards/sound modules, unusual musical devices, equipment reference books and MIDI interface hardware. Hundreds of products are featured that are compatible with most personal computers, including: IBM/compatible, Windows, Macintosh, Amiga, Atari, Commodore 64/128, Apple IIe, IIGS, Laptop, PowerBook, and the newest Intel and PowerPC-based computers."

The price? Well, the press release is a little vague on that. But, it's either $5 or $10, and apparently educators can get it for free. Sound good? Well, here's how to find out more:

Sound Management
P. O. Box 3053
Peabody, MA 01961
Foreign/Technical: (508) 531-6192
USA Sales: (800) 548-4907
FAX: (508) 532-6106

Better Late Than Never

Boy is my face red! I was cleaning out my desk the other day, and I found two booklets that I should have mentioned in this column three issues ago! After a couple of quick calls to the publishers (to make sure that they were still selling the books, and that I had the correct contact and pricing information), I read through both booklets so that I could not only tell you about them, but so I could also give them mini-reviews.

The first book is *The ProLine User's Guide*, which is published and written by Michael D. Porter. As the title suggests, this 51 page booklet is a guide to using a ProLine BBS. As a former ProLine system operator myself, I was amazed at all the stuff that was contained in this book that I had never learned while I was running the old Pro-GSPlus BBS. In addition to covering subjects like the ProLine conference system, mail system, and download file manager, this book also presents a very clear and well-written discussion of the InterNet and its cousins (UUCP, BITNET, and Usenet), and how ProLine boards interact with the InterNet. The book includes several good examples of how to use a ProLine BBS (from logging on your first time to quoting a previous message in a reply), and it also discusses the proper etiquette that users should follow to be good citizens of cyberspace.

Of course, you probably *could* find all of this information (minus the examples) on a ProLine BBS, but you would have to know how to actually get to it before you could get any use out of it. So, if you are a beginning ProLine user, or you are thinking about becoming involved with a ProLine system, you should probably give this book a try. The price is only $12 (which includes shipping costs). For more information, contact:

Michael D. Porter Consulting
231 E. San Fernando St. #4
San Jose, CA 95112
(408) 288-8937

The second booklet is entitled *Exploring SoundSmith* and it's written and published by Al Crout and Gareth Jones. As the title suggests, this booklet is for those folks that want to get more out of Huibert Aalbers' SoundSmith program. The booklet covers everything you need to know, from getting started with SoundSmith all the way to hooking up and using MIDI equipment with SoundSmith. *Exploring SoundSmith* also contains several appendixes that discuss other programs that can enhance the usefulness of SoundSmith. (This book also happens to be an excellent example of IIGS desktop publishing—especially if you are wondering what a StyleWriter can do when hooked to a IIGS.)

The price for *Exploring SoundSmith* is $15, and it is available from:

Gareth Jones
#1, 3872 Sunset St.
Burnaby, British Columbia
Canada V5G 1T3
(604) 435-1416

HyperCard IIGS v1.1

Yes, I know that HyperCard IIGS isn't new, but the ways you can get it certainly are! Early in February, HyperCard IIGS (all six disks of it!) became available for download from most major online services. At this point, I've been able to verify its availability on GEnie, and Delphi, but not America Online or CompuServe. And, if your user group is authorized to distribute the IIGS System Software, they can now distribute HyperCard IIGS too! (Note that only authorized and licensed organizations can distribute HyperCard IIGS in this manner, so don't go putting it on any local bulletin boards. If your user group is not authorized to distribute the System Software, have your user group's president contact Apple Software Licensing at 408-974-4667.) Of course, by downloading HyperCard IIGS or getting it from your user group, you miss out on all of the great (and heavy) documentation, but the Help stack that comes with it is more than enough to get most folks up and running. And, if you really want to get serious about your HyperTalk scripting, you should probably buy the *HyperCard IIGS Script Language Guide* (which is only $24.95 from Resource Central, call them at 913-469-6502 for more information) anyway, so the lack of documentation may not be that big of a drawback. So, if you've always wanted to try your hand at HyperCard, climb online and give it a try!

GEnie
401 North Washington St.
P. O. Box 6403
Rockville, MD 20849-6403
Voice: (800) 638-9636

Delphi
General Videotex Corporation
Three Blackstone St.
Cambridge, MA 02139
Voice: (800) 544-4005

ICONference

At the end of last year's KansasFest, the question everyone was asking was, "How does he get his hair to do that?" Well, I don't know the answer to that one, but I do know that Resource Central has just announced that there is going to be another KansasFest this year! However, since the A2-Central newsletter is no more, this year's show is going to have a new official name: ICONference. This new name refers to the ICON user's group that Resource Central started several months ago (for more information on ICON, see "What's New" in *GS+* V5.N2). However, I suspect that most folks will just keep calling it "KansasFest."

If you want to go to the show, and you should—it's a blast, here's what you need to know: This year's KansasFest will be

held on July 21-23 at the Avilla College in Kansas City. Unlike the last few years, the *entire* show will be held at Avilla, including all of the conferences and the keynote speach. Although final prices have not yet been announced, this change should result in a reduction in the price of admission to the conference.

And, if you want to save even more money, you can get yourself a discount by making a presentation at the conference! While this may sound like a lot of work (and it will be if you want your presentation to be worth a darn), running a presentation at KansasFest is a lot of fun and it's a great opportunity to give something back to the Apple II community. The deadline for presentation ideas is May 1, so, if you want to make a presentation, be sure to contact Resource Central with your idea (send them an outline of your topic, who your intended audience is, and your equipment requirements) as soon as possible.

Before I give you the contact information for Resource Central, let me just say that if you don't plan on going because you are worried that the conference will be over your head, you are dead wrong. Granted, a lot of the material presented will be technical, but if this year is like previous years, there will be several presentations aimed at users and Apple II entrepreneurs. And, you get to hang out for three days with all of the coolest people in the Apple II world—and Bryan Pieterzak too! For more info, contact:

Resource Central
P. O. Box 11250
Overland Park, KS 66207
913-469-6502

Apple FAX

A couple of days ago, a FAX came in from Apple Computer Inc. It was a document catalog from Apple's new "Apple FAX" service. If you aren't familiar with FAX-on-demand services, here's how it works: You call a voice phone number, and, by pressing buttons on your touch-tone phone, you request that a particular document (or documents) be sent to your FAX machine. When you hang up, the service calls your FAX machine, and transmits the documents you requested. It's very cool.

Well, as you might guess, the new Apple FAX service provides you with access to data sheets, price lists, and other information on all currently available Apple products. But it's not all Mac and Newton stuff, there are actually two Apple II products (the IIe card for the Mac LC and System 6) that you can get information on via Apple FAX.

So, get your FAX machine all set up, and call (800) 505-0171 from your touch-tone phone. When you get through, listen to the message, and then request a document catalog. This will give you a listing of all the available documents and their ID numbers so that you can call back and request exactly what you want.

Tell Me About It!

Don't forget boys and girls, if you want us to tell people about your Apple IIGS-related product, service or event, send us a press release! Just send them to any of the e-mail addresses that are on the title page of this magazine, or use the US-Snail and send them to:

GS+ What's New
P. O. Box 15366
Chattanooga, TN 37415-0366 *GS+*

Rumors, Wishes & Blatant Lies

By Prof. G. S. Gumby

Top Ten of All Time!

If you were with us at this time last year, you will remember that in *GS+* V4.N4 we ran the top ten rumors, wishes and blatant lies of all time. However, due to an oversight, we actually printed only five of those ten items. So, this issue, we finish out the top ten rumors, wishes and blatant lies with the top five rumors, wishes and blatant lies of all time! However, they aren't on this page, they are scattered throughout the magazine for you to find as you peruse the other fine reading material in this issue.

By The Way

This isn't a rumor or anything, but since *everybody* reads this column first, it was decided that this would be a good place to mention that our toll-free order number now works from Canada!

Good News!

Did you know that the IIGS version of Out of This World (see review in *GS+* V4.N2, has sold more than 3,000 copies? You didn't? Well then, you probably also didn't know that Interplay was so surprised by these good sales, that they have decided to do another IIGS game in the near future.

Mail Fraud?

It seems that a certain Apple II developer (who shall remain nameless) has filed a mail fraud complaint against Apple Computer Inc. with the Postal Service. Why? Well, according to the developer, even though he (or she) had paid for membership in Apple's Apple II-only developer support program, he was not getting any actual Apple II support At this point, the complaint has supposedly been referred to Apple's legal department, and they are said to be investigating the claim.

Ice, Ice. Baby.

Just when we thought the medication was doing him some good, our publisher got wind of the new "Ice" beers that were introduced during the SuperBowl. This has caused him to resurrect his plans to follow up EGOed light with EGOed Dry, EGOed Genuine Draft, and now, EGOed Ice. (See "Rumors, Wishes & Blatant Lies" in *GS+* V4.N4 for more lies about this.) What's different about these new versions of EGOed? A quick look at the initial specifications for these "new" programs revealed the following:

• EGOed Dry: Title bar of window says "EGOed Dry" instead of "EGOed".

• EGOed Genuine Draft: Title bar of window says "EGOed Genuine Draft" instead of "EGOed".

• EGOed Ice: Window colors are blue and white. Title bar of window still says "EGOed".

Why No Macs?

Speaking of the SuperBowl, if you managed to watch the whole thing, you may have noticed that Apple didn't run any 10th anniversary Macintosh commercials. I wonder why not?

Passport Revoked?

Here's a question that I've been trying to answer for a couple of months now: Whatever happened to *The Passport House Letter*? I've tried calling the publisher and leaving messages, but, so far, no response, and no new issues of the *Letter*. If anyone knows what has happened to them, be sure to let me know!

Oh, I Wish!

What is it that I wish? Well, I wish that someone out there would come up with:

• An MS-DOS FST that would both read and *write* PC disks.

• A program that would let me download packages to my Newton.

• A program that would let me convert .FLI files from the IBM PC to the IIGS.

• A program that would write this column for me (the monkey I've hired is *expensive*).

In The Works

Believe it or not, there are a lot of new IIGS products in the works that should be making their way to market soon. Here's a brief overview of what you can look forward to in the months ahead:

CD-Format - This is another product that allows IIGS owners to make use of CD-ROMs for the Mac and PC. CD-Format is a hardware device that reformats any CD-ROM, erasing all information on the disk. Unfortunately, CD-Format does not include any way to *write* data from your IIGS to the empty CD. When I asked the programmer what good this was, he told me that, "These things make absolutely great coasters, but I always felt guilty when I knew that there was actually data on them. CD-Format helps to remove the guilt."

PowerGS - No, this isn't a IIGS emulator for the PowerPC, it's a new power-strip/surge-suppressor for the IIGS. (Actually, it's just a K-Mart power strip with a "PowerGS" sticker on it. Can you tell that Applied Engineering is trying to get back into the IIGS market?)

SimCity Classic - Yep. Believe it or not, in the next few months, we should be seeing a IIGS version of the incredibly popular game, SimCity Classic! If you don't know what SimCity is, it's a city simulator that puts you in charge of a fictional city. You set the taxes, you build the roads, you break the unions! It's great fun, and educational too!

Ask Mr. 8-Ball

After a brief hiatus (which is much better now, thank you), our outspoken and controversial source, Mr. 8-Ball has returned! Let's get right to those rumors:

Gumby: I understand that our Technical Editor, Joe Wankerl, has been offered the part of Butt-Head in "MTV's Beavis and Butt-Head: The Motion Picture." True?

8-Ball: Yes.

Gumby: Well, good luck to him. Moving back to the IIGS world .... Rumor has it that there has been quite a personnel shake up in the Apple II areas on GEnie? Any truth to this rumor?

8-Ball: Ask again later.

Gumby: Well then, maybe you can give me the skinny on this one. Rumor has it that an Apple II magazine (which shall remain nameless) had to let one of their employees go due to lack of funds to pay him or her. True?

8-Ball: Reply hazy. Try again.

Gumby: OK. Rumor has it that the *real* reason this person was let go was because they were calling "976" numbers on company time. Any truth to that one?

8-Ball: Outlook not so good.

That's all for this time. If you have any IIGS rumors, wishes or blatant lies that you think the Professor should know about, send them to me at:

GS+ Rumors
P. O. Box 15366
Chattanooga, TN 37415-0366 *GS+*

How to Use Your GS+ Disk

The first thing you need to do is **make a backup copy of your *GS+* Disk with the Finder!!!** Do *not* make your backup on your hard disk! Instead, copy the *GS+* Disk to another 3.5-inch disk (this is *very* important). Next, put the original in a safe place. If you are having a problem making a backup copy, give us a call at (615) 843-3988. If your disk is damaged, let us know, and we'll get a new one to you as soon as possible.

Installing The Software

To install the software on this issue's *GS+* Disk, start up your computer using System Software v6.0.1 or later. (Note that all of the programs on this issue's disk [except EGOed lite] *require* System 6.0.1!) Next, place your *backup* copy of the *GS+* Disk in a drive. (You *did* make a backup didn't you?) Now run the Installer program that is on your backup *GS+* Disk. (From the Finder, just double-click on the Installer icon.) *It is* extremely *important that you use the Installer that is on your backup* GS+ *Disk! Do* not *use any other copy of the Installer!*

When the Installer window appears, select the item you want to install from the list on the left-hand side of the window, and the disk you want to install it on from the list on the right-hand side of the window. Then click on the Install button. For more information on using the Installer, refer to your IIGS owner's manual.

Before you attempt to use your backup *GS+* Disk, please take a few minutes to read the **a.Read.Me** file for any last minute corrections or information. If you do not already have our EGOed lite text editor installed in your system, you can use the Teach application supplied with System Software v6.0 to read this file.

Installing EGOed lite

The following is a detailed example of how to install EGOed lite. The other programs are installed in a similar manner.

- Start up your IIGS with System Software v6.0 or later—the version of EGOed lite that is on this *GS+* Disk *requires* System 6! (Your *GS+* Disk is *not* a startup disk, so don't try starting your computer with it.)

- Insert your backup copy of the *GS+* Disk into a drive and run the Installer program that is on your backup *GS+* Disk. It is *very, very* important that you run the Installer that is on your backup *GS+* Disk and *not* some other copy of the Installer.

- When the Installer finishes loading, click on the Disk button on the right-hand side of the Installer window until your startup disk appears. (If you only have one 3.5-inch disk drive, you will have to remove the backup *GS+* Disk from the drive and replace it with your startup disk. You should also refer to the "Making Room" section below for hints on how to free up room on your boot disk.)

Please Remember . . .

The contents of the GS+ Disk are not public domain or shareware! We depend on your honesty to stay in business. Please do not give away copies of the GS+ Disk or any of the programs on it. If you do, we will not be able to stay in business. It really is that simple!

- On the left-hand side of the Installer window, you will see a list of the items on the backup *GS+* Disk. One of the items in this list should be "EGOed lite." (If EGOed lite is *not* in this list, quit the Installer and begin again. Be sure that you are running the copy of the Installer that is on your backup *GS+* Disk!) Once you see the EGOed lite item, click the mouse on it so that it becomes highlighted.

- Click the mouse on the Install button in the middle of the Installer window. The Installer will then install EGOed lite on your startup disk. If you only have one 3.5-inch disk drive, you may have to switch disks several times. Just insert each disk as the Installer asks for it.

- When the Installer has finished, click on the Quit button in the middle of the Installer window. This should cause your IIGS to restart.

- When your IIGS finishes restarting, pull down the Apple menu and select EGOed lite (note that you have to be in a desktop program like the Finder to have access to the Apple menu).

- When it finishes loading, notice that EGOed lite has its own menu bar. Select Open from the *EGOed lite* File menu and then put your *GS+* Disk in a drive. You should see a list of the files and folders on the *GS+* Disk.

- Open the **Documentation** folder on your backup *GS+* Disk and then open the file **EGOed.lite.Docs**. This file contains complete documentation on how to use EGOed lite. *Please take a few minutes to read this documentation.*

Making Room

If you do not have a hard drive, you will probably have to remove some files from your startup disk to make room for the New Desk Accessories, control panels, and other system files on your *GS+* Disk.

Towards that end, we have prepared the following list of "expendable" files that you can "safely" remove from your System Software v6.0.1 startup disk to free up some space. (We've put quotes around "expendable" and "safely" because almost *all* of the files in the IIGS System Software have some sort of use! The files listed here are the ones that are the "least" useful for a specified hardware setup.)

Be sure that you *never* delete *any* files from your original System Software boot disk! Always work on a backup copy!

System Software v6.0.1

If you use the System 6.0.1 **:Install** disk to create a minimal, 800K, System 6.0.1 boot disk, that disk will have 26K free when the installation is finished.

It must be noted that *all* of the files on this disk are *very* important and the files that you can *safely* remove depend, for the most part, on your hardware setup. So, please read these instructions carefully before removing *any* files.

The first two files you can delete depend on what you will be doing with your IIGS. If you will not be running AppleSoft BASIC programs, you can remove the file **BASIC.System** (11K) from the root directory of the disk. If you will not be running ProDOS 8 software, you can remove ***:System:P8** (18K).

If you do not care what time it is, you can delete the following file:

***:System:CDevs:Time** (10K)

After that, the files that you can safely remove depend on your *hardware setup*.

If you have a ROM 01 IIGS, you may delete the file:
***:System:System.Setup:TS3** (42K)

If you have a ROM 03 IIGS, you may delete the following file:
***:System:System.Setup:TS2** (37K)

If you do *not* have a 5.25-inch drive, you may delete the following 8K file:
***:System:Drivers:AppleDisk5.25**

If you do *not* have a printer, you may delete the following file:
***:System:CDevs:Printer** (5K)

Finally, if you have deleted all control panels, and you won't be installing any control panels from the *GS+* Disk, you can also delete the 18K file:
***:System:Desk.Accs:ControlPanel**

Removing some or all of these files will give you ample room (up to 138K on a ROM 01 IIGS and up to 133K on a ROM 03 IIGS) on your startup disk to install EGOed lite or any of the other system utilities from your backup *GS+* Disk.

Having Problems?

If you are having a problem with one of the programs on your GS+ Disk, we want to help! But we can't help if we don't know about it!

If your GS+ Disk is defective, let us know and we will send you a replacement. You can call us at (615) 843-3988 (Monday through Friday between 9 a.m. and 5 p.m. Eastern Time), to request a replacement disk.

If you are having a problem using one of our programs, please fill out the problem form that is on your GS+ Disk and send it to "GS+ Problems" at the address shown below.

Note: You will *not* be able to print from EGOed lite or any other desktop program when using an 800K, System 6.0 boot disk. (There isn't enough room for all of the required drivers and control panels.)

If you want to save even *more* space, you might want to consider using Autopilot (from *GS+* V4.N1) as a replacement program launcher. With Autopilot installed on a minimal System 6.0.1 boot disk, initial free space goes up from 26K to 161K! You can then use Autopilot to autolaunch the Finder from a second 3.5-inch disk drive and still have plenty of room on your boot disk for lots of system extensions. For more information on Autopilot, refer to the "Autopilot v2.0" article in *GS+* V4.N1 or give us a call.

Self-Extracting Archive

We use GS-ShrinkIt v1.1 to compress the *source code* and related files on the *GS+* Disk into a *self-extracting archive*. To extract the files from the archive, simply double-click on the **GSP.V5.N4.SEA** program on your backup *GS+* Disk. ***You do not need to have a copy GS-ShrinkIt in order to use any of the programs or other materials on this GS+ Disk!*** However, you will gain better control over

Use scissors or a knife to open disk bag!
Do not attempt to pull bag away from magazine!

the files you wish to extract if you have GS-ShrinkIt v1.1. If you do not have GS-ShrinkIt v1.1 and you would like a copy, check with your local user group or give us a call here at *GS+* Magazine and we will try and help you locate a copy.

What's On The Disk

There are nine items in the root directory of this disk:

a.Read.Me

A lot can happen from the time we send this magazine to the printer and the time we get ready to mail them out. If anything does happen, we will put everything we can find in this file. Please read this file before using the *GS+* Disk.

Documentation

This folder contains the EGOed lite documentation file, the complete *GS+* Glossary, and some technical notes. The EGOed lite documentation is a Teach file which can be read using Teach, EGOed lite, or any other TextEdit editor. The *GS+* Glossary file is a plain text file containing all of the terms defined in the past installments of the "*GS+* Glossary". The technical notes are also plain text files which can be read by any text editor.

GSP.V5.N4.SEA

This is a self-extracting archive (SEA) containing the source code and related files for all the programs contained on this *GS+* Disk. The archive also contains the Miscellaneous Library. Technical information, such as the Miscellaneous Library documentation is supplied in the archive as well. To extract the files from the archive, simply double-click on this file from the Finder. You will then be presented with a dialog asking you where you want the files extracted to. Note that if you try to extract *all* of the files from this archive at one time, they will *not* fit on an 800K disk!

Icons

This folder contains Finder icons used by the various programs on the *GS+* Disk.

Installer

This is the Apple IIGS Installer. The installer requires System Software v6.0 or later. Run it to install the other programs on this issue's disk. For more information on using the Installer, be sure to read the example on the previous pages, and refer to your owner's manual.

PostScript

This folder contains some sample PostScript® programs. You can use LASERbeam to send these programs to a PostScript laser printer. Be sure to check the **a.Read.Me** file in this folder for information on what each program does.

Programs

This folder contains the EGOed lite, LASERbeam, Playful, and What Is This? programs. Use the Installer provided on your backup *GS+* Disk to automate the installation of these files. EGOed lite requires System 6 to operate. All the other programs on this disk require System 6.0.1 to operate.

Scripts

This folder contains all of the scripts that are used by the Installer to install the files from this *GS+* Disk.

Talk.To.GSPlus

This folder contains our feedback form, a troubleshooting guide, a problem form, and our writer's guide. The feedback form is a plain ASCII text file. Fill it out, and let us know what you thought of this issue. The troubleshooting guide contains tips on how to resolve some of the more common problems you may experience while trying to use the programs on your *GS+* Disk. If you are having a problem, *please* read this file before you go to all the trouble of filling out a problem form! But, if the troubleshooting tips don't help, *please* fill out the problem form and send it to us! This is a Teach file, you may use EGOed lite or the Teach application to view it. The writer's guide is a Teach file that explains what you need to know to write for *GS+* Magazine—you may use EGOed lite or the Teach application to view it. *GS+*

How to Get System 6.0.1

Everyone should have a copy of System 6.0.1. Fortunately, we have a license to distribute it to our magazine-and-disk subscribers as a part of their subscription. Unfortunately, we can't afford to mail all five of the disks that System 6.0.1 takes up to every magazine-and-disk subscriber. However, we still want to make it easy for you to get System 6.0.1. So, if you are a subscriber to *GS+* Magazine with the companion *GS+* Disk (sorry, but we can *not* distribute System 6.0.1 to our magazine-only subscribers), send us the following items and we will send you System 6.0.1:

1) Five (5) *blank* and *formatted*, 3.5-inch diskettes to our P. O. Box address (which is shown on the back of your magazine). We are asking for "blank and formatted" disks because formatting takes time that we don't have, and it's a great way to tell if a disk is good before you send it to us. *If you send us a bad disk, we aren't going to replace it.*

2) A *self-addressed* return disk mailer with enough postage on it to mail the five disks back to you. (Foreign subscribers without access to United States postage may include International Postal Coupons instead. See your local post office to obtain these.) *If you don't provide a postage-paid, self-addressed return mailer, your disks will be considered "gifts" and will be used for backups.*

3) That's all. Don't send any money. We don't want any money for this.

How Else Can You Get It?

If you are a magazine-only subscriber, here are some other ways to get System 6.0.1.

Your Apple dealer. Bug them until they get it in for you. The retail price is $39, but that includes manuals. The part number is #A0077LL/A. For the name of your local Apple dealer, call (800) 538-9696.

Your user group. Take your own disks and they should only charge you a small copying fee. Some user groups may have it already copied for you and available for a nominal charge. (Note that some user groups make these services available only to their members. Of course, you do plan on joining, don't you?) If you need to know where your local user group is, call the Apple User Group Connection at (800) 538-9696 extension 500.

Resource Central. You won't have to bug them, they have it in stock, and in no less than two different "flavors." For just the disks (item number DA-006), the price is $24. For the complete end-user package, including manuals, the price is $39 (item number DA-0013). Take your pick, and then give Resource Central a call at (913) 469-6502.

And, of course, if you have a modem, you can download it from your favorite online service. The total download time is about 5 hours. *GS+*

Reviews

Addressed for Success

Written by Michael Lutynski

Typical mail order price: $49.95

Not copy protected
Requires System Software v6.0 or later, 1.5MB RAM or more, and a printer with appropriate driver. A hard disk and Avery brand labels are recommended. Complete installation takes up 434K on disk.

Econ Technologies Inc.
P. O. Box 195356
Winter Springs, FL 32719
(407) 365-4209

Reviewed by Ron Hochevar

Addressed for Success (AFS) is a new address database and label printing program written specifically for the Apple IIGS. It employs the familiar desktop interface and does an excellent job of implementing many of the features unique to System 6. At the time I initially agreed to write this review for *GS+* Magazine, I felt that it would be a short review about a simple label program. However, as I started to further explore the many features that this program has to offer, I realized that it would be impossible to briefly explain them all. It took me a moderate amount of time to become familiar with its operation, due to the fact it allows for a nearly limitless number of variations to the final output. It became a challenge as I attempted to perform increasingly detailed graphical layouts to complement the great looking text. I found myself cranking out labels left and right in my attempt to find just the perfect combination of graphics and type style in my pursuit of the perfect label! One of the by-products of this venture is that I now possess several years worth of return address labels for every member of my family!

Econ advertises this program as a "desktop management package that lets you manage large lists of names & addresses." While this is certainly true, many average IIGS owners that make labels on a limited basis (such as return address labels and labels used for special occasions such as Christmas cards) might wonder whether or not it is worth the investment, especially if they already own an existing program such as Big Red's "Labels, Labels, Labels," Broderbund's "Print Shop Companion" or use an integrated program such as AppleWork's GS to create their labels. This may be a difficult decision to make, but hopefully the information contained in this review will give some clues as to whether or not this program is likely to fit your needs. I decided to purchase the program because I own a Hewlett-Packard DeskJet 550C (in addition to an ImageWriter II). I wanted a label-specific program that would print color labels at the DeskJet's 300 d.p.i. resolution using Vitesse's Harmonie HP printer drivers along with TrueType fonts from WestCode's "Pointless" package. (As I am sure most IIGS owners already know, a desktop-based program is a necessity to use these drivers and fonts.) Before I go any further, let me point out that I am not a techie or a programmer. Throughout this review I make suggestions and criticisms of this program, some of which may be, from a programming aspect, impossible to implement. However, this is not going to prevent me, as the saying goes, from "throwing in my two cents worth" concerning how I think it could be improved. With that being said, let's get on with the review . . . .

The Manual:

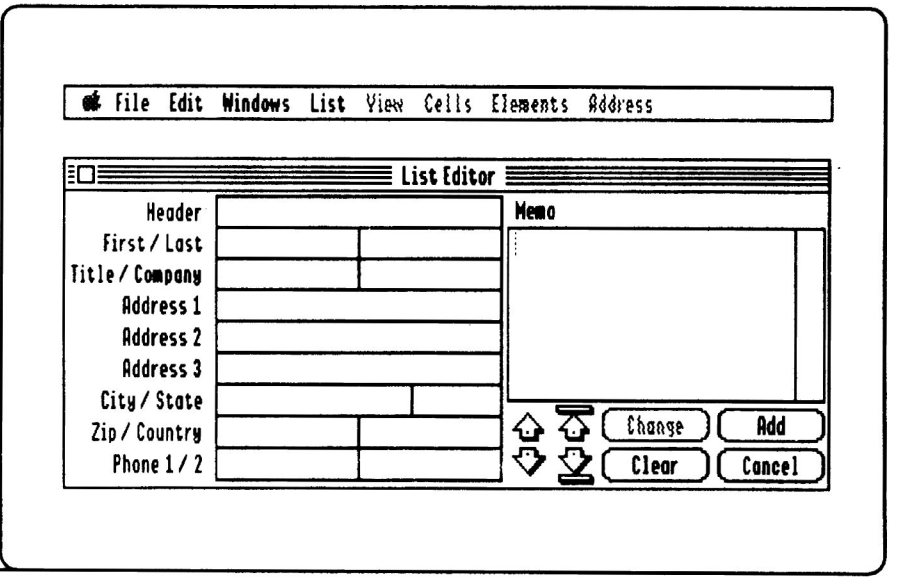
Econ supplies a 35 page manual with the program that begins with a brief explanation of what AFS is designed to do and the proper method of installing the program for single disk drive users, dual floppy drive users and hard disk owners. While AFS can technically be run off a single 3.5-inch floppy drive, because of the nature of the program and extensive use of the IIGS's clipboard when importing graphics, a hard disk, in my opinion, is almost a necessity. If you should decide to use it with floppy drives, put your patience cap on before booting the program!

After installation, the manual goes on to explain the basic terminology and features incorporated into the program, followed by a section describing an overview of the options found in the pull-down menus. A three part tutorial familiarizing the user with the operation of the program's features is next followed by a reference section describing each individual menu function in greater detail. The final portion of the manual deals with troubleshooting problems that might occur and how they can be corrected or avoided.

I found the manual to be well written, easy to understand and not overly wordy in its attempt to explain how to successfully use AFS. This is a particularly noteworthy accomplishment since it introduces some features and design concepts that are totally unique to AFS and are not likely to have been encountered by the user with other programs. (I found the manual good enough that I decided to use it for the outline of this review.) However, it does have some major omissions, which I will discuss in detail later in this article. So let's get on to the "Nuts 'n Bolts" of AFS . . . .

The "List" file

AFS employs the use of two distinct types of files that are combined to create the final label output. The first, the list file, deals with the creation and manipulation of the actual address data. Two methods can be used to create the list



file. Method one entails the use of the "List Editor" window where address information is typed in for each record (see first screen shot) The editor contains 15 separate fields that can be used to fill in pertinent information. These fields include: Header, First Name, Last Name, Title, Company, Address 1, Address 2, Address 3, City, State, Zip, Country, Phone 1, Phone 2, and finally "Memo". Any or all fields, with the exception of the memo field, may be printed. As a practical matter though, one would have to select a rather large label and/or extremely small type size in order to get all 14 fields printed. Each field may contain as many as 200 characters with the memo field holding as many as 65,000 characters.

The second method of data entry allows the user to import Classic AppleWork's database or ASCII text files (formatted in two different ways). I had no trouble importing AppleWork's database files using the special dialogue box that allows you to match fields for importation, however, my attempts at importing ASCII text files with carriage returns between fields and records proved fruitless . . . it just did not work correctly. All of the data ended up in one big bunch under the name field. I contacted Econ about this problem and they are in the process of checking into it and making a correction to the program if necessary. A second method of importing ASCII text files, those with tabs between fields and carriage returns between records worked fine. [As a sidelight to this method, I was successful in importing data files from the telephone dialer/database program "Contacts GS" written by Bill Heineman and published by Simplexity Software (See review in *GS+* issue V4.N1) with only minor editing required prior to importing.]

The entire contents of the list can be viewed in what is termed the "list window." This scrollable window displays the list name, fields, records and whether or not records have been selected for printing. Fields may be arranged in any order by sorting. A postal "bulk sort" can also be performed when the zip code is arranged as the first sort field. Records denoted with a check mark in front of them are marked as printable. You may also select a single address as the return address from within the list. It then shows up in the color blue. Later, when in the layout section of the program, you can either have the return address printed somewhere within your outgoing labels (if they are large enough) or make separate return address labels automatically, matching the exact number of labels you are sending out . . . another nice feature of AFS!

AFS employs two search methods to locate specific records or groups of records within your list. The first is mainly used to alter your data within the list. The Find option under the edit menu brings up a dialog where you can search for specific records that contain user defined strings. It then allows you to edit the fields by bringing up the list editor if so desired.

The second method is used to determine just which records you want to have printed out. Selecting the Find By Example option, (also under the Edit menu) allows you to enter single or multiple criteria for locating those records that you want to select or deselect for printing. Any of the 15 available fields may be used to conduct the search. An example of this might be to search the list for all records containing the name "Rick" in the first name field and "WA" in the state field. This would cause all those records that match to show up in inverse in the main list window. From there you would mark them as printable using that option via the List window menu. Wildcards may be used when conducting a search, which is handy if you are not sure of the exact criteria to search for. For instance, in the above example, commencing a search using the wildcard characters (denoted by an asterisk "*") for *Ri* would bring up every record with the letters "Ri" in them . . . thus, you would not have to remember if the person you are searching for is named Rick, Richard, or Ricardo, they would all be selected by the program for you. As mentioned, this can be handy and the manual explains this feature very well. A third option allows you to select records

manually by holding down the Command (Open-Apple) key and clicking on each entry as you scroll through the list or use the shift-click method of selecting consecutive records. This puts them in inverse to be marked as either printable or taken off the print list (unchecked). I had no particular problems executing these functions. However, a couple of things could be done to improve this area of the program.

First, having the capability of importing AppleWorks GS database files directly into the program using a method similar to that used with Classic AppleWork's files would be a nice addition to AFS. While there is no direct means of importing these files, they can be imported if you first export these files out of AWGS in ASCII form. Be sure to export only those fields that you wish to import into AFS and in the field order in which they will be imported.

The manual makes no mention how one goes about making a simple print out of the records (either all or in part) of the database. While I initially thought this would require the rather lengthy process of exporting them out first in ASCII format to load into a word processor, I was able to find a much quicker method. It can be found in the "Tips For Using AFS" section at the end of this review.

While it did take a little time to become familiar with this portion of the program, I found the list creating, editing, and marking features easy to use and quite powerful in allowing you to manipulate the data quickly and easily.

The Graphical Element . . .

The second type of file that is used within AFS is called the "layout file." This portion of the program is where things really get interesting! Most folks use some size of Avery brand labels on which to print. AFS comes with 32 pre-sized Avery label layouts specific for the ImageWriter II and LaserWriter printers (16 for each printer). Simply select the correct label number that corresponds to the number on the package of the Avery labels you wish to use and you're in business!

At this point, it is important that you specify your printer, printer driver, the paper type and aspect ratio for the type of label that you will be using. This information affects how the layout file appears on screen, and is saved when you save your layout file. Your layout file is then brought up on screen in "What You See Is What You Get" (WYSIWYG) format, which means you get a view of the layout file as it will be printed! A graphical representation of the edges of the paper is displayed in red, the border of the printable surface in blue, and the actual label outline in green (see second screen shot). Only those records marked as printable in the list file are displayed within the individual layout "cells" (cells are graphical representations of the individual labels). Any combination of fields contained in the list file may be chosen from the Address menu and displayed on screen. For instance, if you had all 14 printable fields filled in on your list but only chose to print the name, address, city, state and zip, you could select only these items under this menu to be displayed and printed. Individual attributes that are user definable within the layout include: type of font, style and size, label dimensions as well as margins. An important and unique feature of the layout file is that they are interchangeable, that is, any layout file can be combined with any list file to create the final output. This makes the program extremely flexible. Once you begin accumulating saved layouts, you can quickly mix and match graphics from one or more of them via the copy and paste method, creating a whole new layout in the process!

Elements

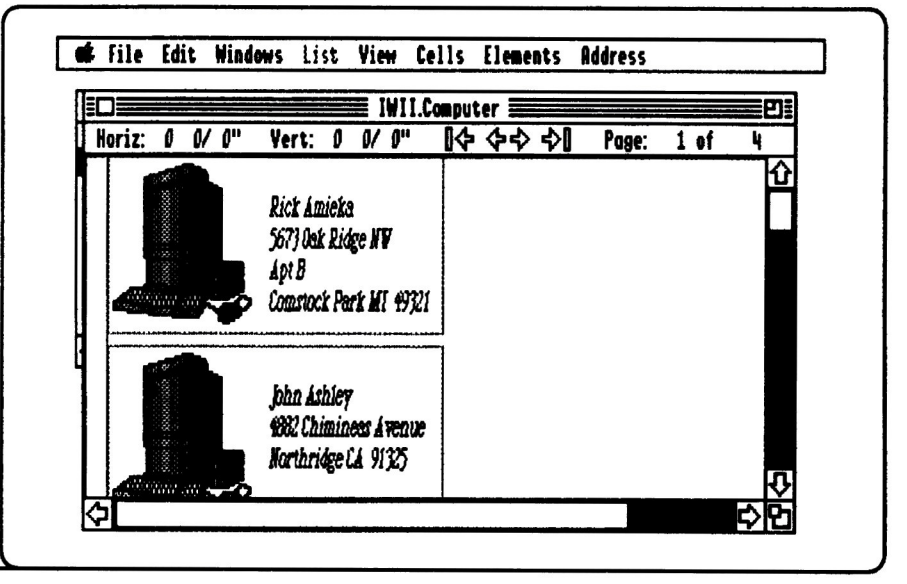
Four distinct components make up what are termed the "elements" of a layout file. These are: Graphics (up to three per cell), the address, return address and postal bar code. A nice feature of the program is that it allows you to move these elements around at will. For instance, if you find the text is too close to the graphic, simply click on the text element within any cell to select it and a box appears around the text allowing you to move it. Releasing the mouse button sets the position of the text for not only the individual element you had selected, but for all of the text elements in the layout. The same is true for any of the other elements mentioned above. This allows the user to quickly make adjustments to their layout. However, there is downside to this. If you have records that contain long address lengths mixed in with records that have short address lengths, you may have to compromise in the positioning of the text in order to center them onto the labels at one printing. This is especially true if you are attempting to place a border graphic around the text. A way around this is to print only those addresses that are of similar lengths at one time.

It would be nice if a feature was added to the program where you could freeze or lock elements within individual cells while leaving the rest free to manipulate. I'm not sure that this is possible given the overall nature of this part of the program, but it would allow for even greater flexibility.

Importing Graphics

This is one area where the manual causes a lot of confusion, (Econ is aware of this and is going to revise this portion of it in future versions). My main gripes and suggestions for improvement deal with this area of the program.

In order to add a graphic element, it must be pasted into the layout via the clipboard. The graphic has to have been placed on the clipboard using: (1) an outside source (such as a paint program) or (2) closing the current layout, opening up an existing layout that already contains a graphic, copying it, closing that layout, reopening the original layout, and finally pasting it in. If this sounds a bit confusing, it is . . . at least to the new user. What made it so confusing is that Econ chose to use the latter method for the tutorial. This may be acceptable for



demonstrating the "copy and paste" procedure but will not work when the uninitiated go looking for their first "non-tutorial" graphic! Method one, which, out of necessity, will be used most of the time, is barely mentioned. Also, the manual never really discusses how to properly use the clipboard, which I believe deserves at least an entire chapter of discussion. As an example of this, no mention is made of the likely results if 320-mode graphics are imported into the program. (AFS operates in 640-mode, causing the colors of the graphic to change to the 640-mode palette. This in turn, will more than likely, cause the graphic to look pretty awful when pasted.)

I used standard 3 1/2" x 1 15/16" labels for nearly all of the testing on this program and found that because of this, the size of the graphics must be pretty small to fit the label. Larger graphics that had to be resized to a smaller dimension often lost too much resolution to be acceptable.

I'm not much of an artist so I decided to use Print Shop GS graphics, which I found to be in most cases, suitable for the purpose. I was able to cut and paste successfully using Platinum Paint's import graphics option to first load the PSGS graphic into it, make adjustments to the picture, copy it back to the clipboard and then run AFS to paste into my layout. This is a time consuming process. It would be nice if AFS would allow for direct importation to the clipboard of PSGS graphics from within the program. However, there is a way to get around all of the bottle necks presented with this method. There are several good scrapbook type new desk accessories (NDAs) around. (Check the major on-line services, user groups or your local Apple BBS to locate one. My personal favorite is "Scrapmaster," by David Huang. It comes with the program SwitchIt from Procyon) These NDAs allow you to store multiple pieces of clip art or other graphics (imported via the clipboard) and then copy them back individually onto the clipboard at a later time. You can then simply paste them into your AFS layouts without having to bounce back and forth from AFS to your paint program! The "Clip It" NDA (freeware by Mike Nuzzi) is also very helpful when you want to clip a graphic from a desktop program and place it onto the clipboard. It would have been very helpful if these options had been discussed in the manual or these NDAs included in the AFS package.

For the ultimate ease of use and flexibility, I suggest using AFS with a multi-tasking type of program. I tried AFS with both SwitchIt from Procyon and The Manager from Seven Hills Software, and found them both compatible. SwitchIt was considerably quicker when redrawing the screen and a bit more stable overall. These programs allow you to change between applications (i.e. AFS and Platinum Paint) without having to back to the Finder. This is a significant time saver when loading graphics to the clipboard to use in subsequent pasting to your layout.

After pasting the graphic, you can further change the way it looks by selecting the Scale Pictures option. A window pops up that allows you to either decrease or increase the horizontal and vertical size of the graphic and to preview the change before applying it back into the layout! This is an exceptional feature that I found both unique and easy to use. You can use this tool to not only resize standard graphics but also custom borders, lines or anything else that you can copy to the clipboard. As mentioned, up to 3 graphic elements can be added to each layout. I used this feature to scale pictures that I had previously digitized and converted using the Visionary GS video digitizer and those created with the Quickie hand scanner. With a little practice, I was able to come up with what I consider to be very interesting and unique labels!

Graphics that may be imported can come from any number of sources. Basically, any 640-mode program that allows you copy to the clipboard directly or gives access to the Apple menu to use a scrapbook NDA to copy to the clipboard will work. Converting 320-mode graphics or clip art to 640-mode and then copying to the clipboard is also an option, however, many times these graphics must be edited prior to copying, normally via the use of a paint program such as Platinum Paint.

While working on this review I discovered a great little utility by Eric Shepherd called "Greyscale Init". Basically, this init sets your default desktop color to the shades of grey used by the Quickie hand scanner. This made it quite easy to get Quickie 640-mode scanned images into AFS in their original greyscale palette without any cumbersome color conversion! This is one of the best little utilities that I have come across. It is freeware and can be obtained via Genie or America Online.

One of the nicest features of the layout portion of the program is found under the Cells pull-down menu. It is the Clip Elements To Boundaries option. By turning this option on, you can quickly see how the graphics and text will fit

Sample Labels

GS'BS BBS (Open 24hrs)
"Bungler" Sysop
1200-14.4 bps / 8-N-1
Bloomington CA
(909)369-6637

Ron Hochevar
Computist, Reviewer &
Apple IIGS Fanatic
Bloomington CA

A-1 Computing Service
17452 Marygold Blvd
Portland OR 86541

GS+ Magazine
Steven "Dis" Disbrow
Publisher
P.O. Box 15366
Chattanooga, TN 37343-15
(800) 662-3634

within the green outline denoting your label cells and allows you to view the individual elements exactly the way they will print out. This is where you can really become creative with your layouts. You can hide elements temporarily by moving them completely off your cell layout and then locate them once again by turning this option to off. In this manner, you can try a limitless number of variations to your labels and see how they will look prior to actually printing them out. This is one of the features that sets this program apart from other label making programs I have used in the past. I found myself varying the text styles and types, mixing and matching different graphics, adding borders, lines, etc. This is one of several features that makes AFS so flexible and fun to use.

The final major component that makes this program so unique is again found under the Cells pull-down menu. As you recall, I mentioned at the beginning of this article that I wanted to use the program with a DeskJet printer. The pre-made layouts are for use with either the ImageWriter II or LaserWriter printers but no DeskJets . . . this turned out to be no problem at all.

Selecting the Set Cell Dimensions option brings up a window where you can define the exact dimensions of the label you wish to use. I had purchased some Avery address labels (#8160) that were specifically designed for inkjet printers. As the manual explains, by measuring the exact size of the labels and inputting the dimensions into this window allows one to create any size layout template for any size label!

Another option under this menu allows you to easily define the left and top margins of your label. The result of all of this is that your label layout as defined in these options will be drawn in WYSIWYG format on your layout screen. A very nice feature indeed! Several other nice features that should be mentioned: the ability to view the layout in several different magnifications (25, 50, 100, 200 and 400%) of actual size. This is helpful in placing your elements exactly where you want them. A horizontal and vertical indicator denoting increments in inches as the cursor moves in the layout window . . . this again is helpful in placing your elements. The ability to jump back and forth between your list and layout windows by either clicking on inactive window or by selection via the Windows menu makes moving between them a breeze. Several arrow buttons make it easy to jump between pages in your layout file or while in the list editor. All of these features and a few that I have not mentioned make AFS quick and easy to use once you become familiar with the program.

The Final Output . . .

When testing this program, I used two separate drivers for the ImageWriter II. Initial testing was performed with the Apple standard driver that comes with System 6.0.1. I had excellent results with no problems. Unfortunately, I did encounter some difficulties with the program when using the ImageWriter II drivers supplied with Vitesse's version 2.0 Harmonie package. I found the WYSIWYG display flawed with the A4 letter (12" long) page format selected and standard 3 1/2" x 15/16" pinfeed labels setup in the cells menu (Avery #4145). The printable surface denoted by the blue rectangle would not match the actual label size using normal or condensed aspect mode in page setup! I spent many hours and a myriad of different settings, both within the page setup and in the program itself trying to come up with some sort of reliable method that would layout and print correctly in a redundant manner. I finally managed to accomplish this (see "Tips For Using AFS" following this review) and was able to produce some excellent labels. Hopefully the author can work on this problem and make adjustments to the program to accommodate these drivers in future updates.

After a bit of trial and error using the DeskJet (and the Harmonie drivers for it), I was again successful in my attempt to consistently produce superior labels. I did experience a few minor problems (see below) but overall, I have been very pleased with the AFS/Harmonie/DeskJet combination.

Random Gripes:

Although the manual discusses the correct settings for the page setup when using the Apple ImageWriter II and LaserWriter printers and Apple drivers, no other specific printers/driver setups were mentioned. It would have been nice if information had been supplied on more printer types. For instance, through trial and error, I discovered that when using the DeskJet, any aspect ratio other than Best caused the DeskJet to print graphics in an unpredictable and often-times strange manner . . . such as one or more columns of graphics printing in the wrong color or containing a vertical bar in back of them. While it would be impractical to expect the setup information to be supplied on every possible type of printer/driver combination, it would have been nice if the manual had included at least the half dozen or so most often used printers and driver setups. If not in the manual itself, perhaps a simple "read me" type file with the appropriate information would have sufficed . . . I know it would have saved me a lot of time and effort!

Econ's advertisement for the program mentions AFS's ability to print envelopes. While I suspect that through trial and error this might be accomplished with the program, no mention of this feature is contained within the manual. This left me wondering whether the advertising was correct or this feature was left out of the program.

There is presently no way to colorize the text in the address. I'd like to see this option added to AFS.

It would be nice to have the ability to use the cursor (perhaps in conjunction with a modifier key) to move the underlying text when a border graphic has been placed over it. This would make centering it within the border graphic a bit easier.

In Conclusion:

I would describe this program as being extremely flexible and provides the user the tools necessary for creating unique looking labels of every size and variety. If you need a program to print just text on labels and a way to manage the data that goes on them, then I would highly recommend it. If, however, text and graphics is your intent, I might have some reservations unless your system has enough memory to include the add on NDA's that I mentioned above. To make it work to it's utmost speed and ease of use, I'd recommend either of the multi-tasking programs mentioned in this article. Although it may seem that I found fault with many areas of the program, this is really not the case. I think I did however, find quite a few areas where it can be improved.

While this program is certainly more complex than other label making programs I have used and takes a bit longer to master, it more than makes up for this by allowing a much greater degree of flexibility and creativity in defining your labels. I believe this is an extremely powerful first version of AFS. I give a lot of credit to to the author, Mr. Lutynski for his considerable effort in creating Addressed for Success. I hope that the few problems mentioned in this review will be worked out, the manual revised to be a little more comprehensive, and if possible, some of the suggestions contained in this article incorporated into future versions. *GS+*

Tips For Using AFS

Although I do not consider myself an expert in the use of Addressed for Success, I did manage to spend a great deal of time working with the program while in the process of writing this review. In so doing, I managed to discover a number of things that can be done that I thought might be of interest to the readers of *GS+* Magazine. Hopefully some of the following tidbits of information will make using the program easier for current or future owners of the program.

• If you want to make a quick print out of any records contained in the list window, select them with the mouse (causing them to be highlighted) and copy them to the clipboard. Open up a text editor NDA such as EGOed from *GS+* Magazine and select New from the file menu, then select Paste from the Edit menu to bring in the records to your document. Finally, select Print from under the EGOed File menu. While the field names will be absent, this is usually no big deal . . . all of the information contained in the records will be available on the print out.

• Be sure to check the clipboard after copying your graphic from within your paint program. If the graphic has changed color from the original, use your lasso or marquee tool to move the graphic in the paint program and then re-copy to the clipboard and check again. This problem occurs in 640 dithered mode and has to do with the odd and even pixel arrangement. I've sometimes had to move and re-copy to the clipboard several times to get the color to stay identical to the original.

• When using the Harmonie ImageWriter II driver and 3 1/2" x 1 15/16" continuous pinfeed labels (Avery #4145), I found the following page setup to work well. Select A4 for page size and ImageWriter condensed aspect ratio. Use the first cell (uppermost) as the template for all of your cells. Anything that can be fit within the blue and green rectangles of this cell will print correctly on all subsequent labels.

• Never try to "back out" pin-feed labels with the ImageWriter II. Push the select button on the printer to turn off that light and then advance the labels through the printer using the form feed button. When loading labels, tear off the amount that you are likely to use before loading.

• If you find you need to adjust the labels to get an even upper and lower margin while the printer is printing on an ImageWriter II try turning off the select light (by again pushing in the select button) just before it begins printing the next label, adjust the carriage up or down and then push the select button again to continue printing. If you are not quick enough this may cause one label to be a little messed up, but better one than all of them!

• You can also increase the size of a graphic element to make it *larger* than 100% of the original size using the Scale Pictures option and typing in the amount (for instance "150"). To get the best results, adjust the ratio for both the vertical and horizontal to the same amount (or close to it). In other words, if you need to increase the size of the graphic by 10%, add this amount to both vertical and horizontal. This will keep the graphic's dimensions uniform with the original.

• When scaling graphics, you may find that certain percentages cause a line to show up in rear of the picture after hitting the preview button. If not corrected, this line will also show up on your labels (assuming you apply it to your layout). This is again due to the dithering problem in 640 mode. I found that increasing or decreasing the horizontal size by one percent will make the bar dissapear.

• When using "The Manager" with AFS, the printer setup dialog did not completely redraw. The options were still available and can be accessed by blindly clicking on different areas of the almost blank dialog window. I don't know what causes this but it happens with regularity.

• Sometimes, pasting more than one graphic into the layout window will cause the last graphic to "stick" to one of the elements already in the layout. I found that clicking on several different elements and moving them around freed up the new graphic.

• Frequently after looking at a printed label I would decide I wanted to make exact adjustments to the positioning of one or more elements. I found it helpful to take a ruler and measure from the vertical and horizontal edge of the label and use the cursor increment feature to re-position the elements exactly where I thought they would look good. This method seems to work really well.

• For my final suggestion . . . I found that when using the ImageWriter II and pinfeed labels (#4145's) I could do test runs without having to burn up a complete set of 12 labels (the usual number per page) by setting the number of rows to two or three in the Set Cell Dimensions option under the Cells menu. The printer will issue a form feed after printing and you will have to feed your labels back in but this works great for testing purposes. *GS+*

Professor Gumby's Top Ten Rumors of All Time
Number 2 - From *GS+* V2.N3
January-February 1991

The Next Generation

Co-processors and parallel processing are two of the hottest technologies in computing today. In an effort to bring this technology to the IIGS, Applied Engineering is said to be working on a new product that will "Make Macintosh owners rip their spleens out with envy!" This new product, which will be called "The Borg" is said to be a large silver box that contains literally *thousands* of processors. Each of these processors will work in concert with the IIGS CPU to complete calculations in an almost instantaneous fashion. There will also be several thousand backup processors that will take over if one of the main processors malfunctions.

According to an AE spokesman, "We had one up and running for a couple of hours. It was amazing! We actually had *four* documents open in AppleWorks GS! At one time!!!" Unbelievable? It sure is! When can you expect to get your own Borg box? According to the same spokesperson, "Well, everything was going great. We were really happy with the tests we had done, so we turned the thing off to break for lunch—and it exploded! I'm not really *sure*, but I think the Quality Control people are going to want us to fix that before we ship, so it may be quite a while."

Creepy Crawly Debuggers (or ORCA/Debugger vs. Splat!)

ORCA/Debugger

By Mike Westerfield
Retail price: $49.95
Not copy protected
Requires System Software v5.0.4 or later (System 6 recommended), ORCA/C, ORCA/Pascal, or ORCA/Modula-2, and 30K of disk space.

The Byte Works, Inc.
4700 Irving Blvd. NW, Suite 207
Albuquerque, NM 87114
(505) 898-8183

Splat!

By Michael Hackett
Retail price: $59.95
Not copy protected
Requires System Software v5.0.4 or later (System 6 recommended), ORCA/C, ORCA/Pascal, or ORCA/Modula-2, and 92K of disk space.

Procyon Enterprises, Inc.
P.O. Box 620334
Littleton, CO 80162-0334
(303) 781-3273

Reviewed by Josef W. Wankerl

Buck Up Lil Campers!

Once upon a time, in the land of IIGS program development, there lived a couple of high level languages—their names were ORCA/C and ORCA/Pascal. Programmers were happy that they could choose the language they knew best to work with. And, to help them debug their programs, there was the benevolent GSBug machine language debugger. Although GSBug did allow the nice programmers to view their code as it executed and help to identify errors, it did so in a manner which wasn't very intuitive to high level language programmers. (Would you rather look at "`x = y`" or "`lda $4332, sta $3FD0`" ???) One day, a new debugger, ORCA/Debugger, became available which allowed you to view your actual high level language source code as it executed (see the review in *GS+* V4.N1). Programmers rejoiced! A newfound glow of peace and tranquility reigned over the land of development.

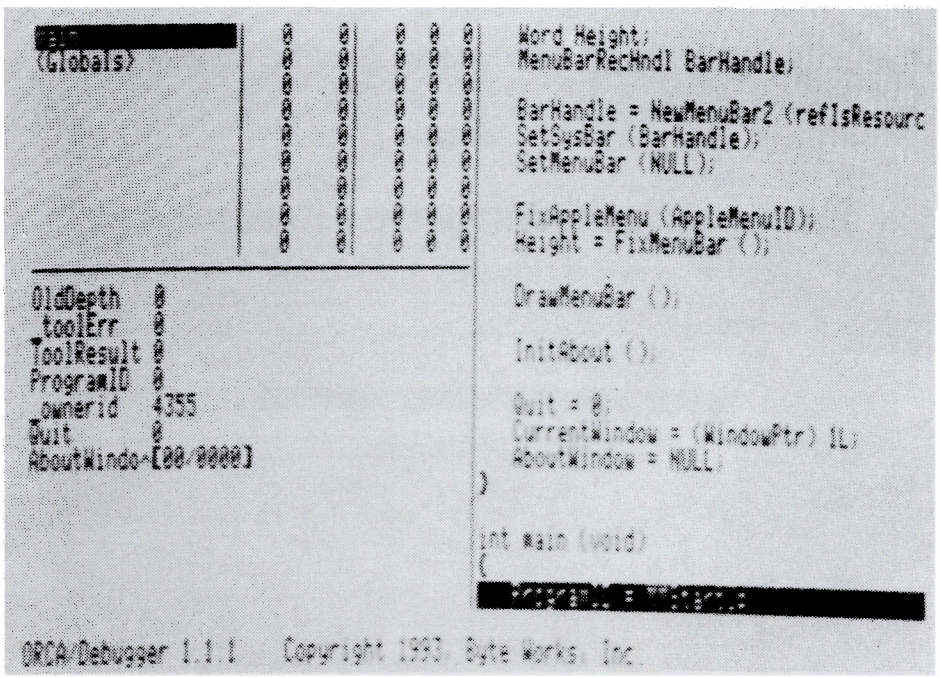
Alas, as things often happen, the two high level languages were changed (for the better), a new language, Modula-2, was introduced, and the ORCA/Debugger could no longer understand the new and improved debugging information that the compilers were generating. The land of development fell back into a well of darkness. With GSBug as the programmers' only recourse, debugging code generated by high level languages became a tedious, evil chore.

Much time passed and much hope for the future was lost. But, when things can't get any worse, they have to get better. Such was the case for the high level language programmers. Not one, but *two* high level language debuggers able to decode the new debugging instructions have finally emerged into the land of programmers! The old ORCA/Debugger was revised to handle the new debugging information and a new program, Splat!, entered the market.

Happy! Happy! Happy!

When I wrote Balloon for *GS+* V5.N2, I spent a *lot* of time debugging it with GSBug. I wasn't very happy at the fact that ORCA/C wasn't compatible with the ORCA/Debugger. I could have gotten the program finished a lot sooner if I'd been able to actually see what my C code was doing instead of trying to figure out what the compiled code was doing and why. In order to see what I had missed while debugging Balloon, I decided to put both the Splat! and ORCA/Debugger programs through some paces.

The first thing I did after I installed Splat! was to recompile Balloon with debugging code turned on. I then installed Splat! and rebooted my system. After everything had started up, I attempted to extract the contents of a ShrinkIt archive from the Finder using Balloon. Lo And Behold! In popped the Splat! debugger and in front of my eyes was the source code ready to execute. I timidly pressed the space bar a few times and watched each line of code go by, savoring each moment. I was watching my code execute! I could see the changes being



ORCA/Debugger

made to my variables! Splat! had passed its first test with flying colors.

Next I pulled Splat! out and popped in the new version of the ORCA/Debugger. I entered the Finder and attempted, yet again, to extract the contents of a ShrinkIt archive. Lo And Behold! Nothing happened. Oops, I had forgotten to turn on debugging breaks. Splat!'s default mode is to break on the first line of code where debug information is present. The ORCA/Debugger's default mode is to hide in the background. So I turned debugging breaks on and tried again. Lo And Behold! In popped the debugger and in front of my eyes was what appeared to be *half* of my source code—the left half to be exact. Yes, the ORCA/Debugger interface was still the same by only showing the left half of the source code. I could step through the code fine and see changes being made to my variables, but it wasn't nearly as elegant as the user interface that Splat! presents.

Next I wanted to debug the LASERbeam program that I updated for this issue. However, the ORCA/Pascal compiler *refused* to generate debug code! Every time I turned on debug code, the compiler spat out a generic "compiler error." When debug code was turned off, everything went fine. Needless to say, I spent a lot of LASERbeam development time in good old GSBug and a lot of time complaining about the compiler. That's a lot of good debugging time wasted.

Ease Of Use

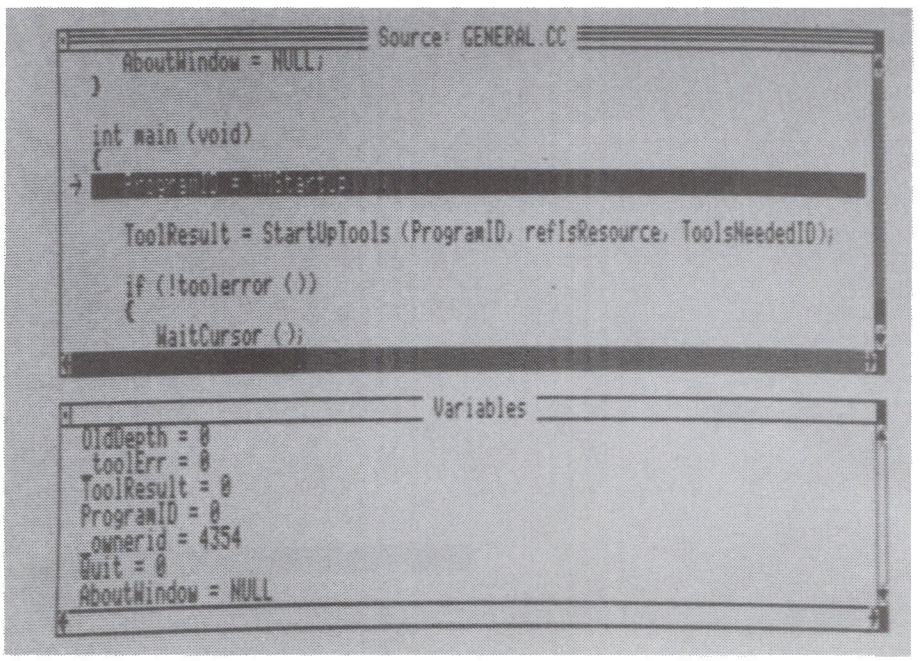
After reading through both the ORCA/Debugger and the Splat! manuals, I set out to do some debugging. Both debuggers will let you step and trace through your code, and both will let you change the values of your variables (even if they're in a record or structure). I found myself testing Splat! a lot more than the ORCA/Debugger. There are two reasons for this: the first is that I had already used the ORCA/Debugger back before the new compilers came out and knew what it could do, and the second is that I just liked the way Splat! worked. The help screens are easy to follow and it's not difficult to see what's going on.

To make my point about how much easier Splat! is to use than the ORCA/Debugger, I'll tell you how they both implement auto-go ranges and break points. First let's set a break point in the ORCA/Debugger. On the command line, type BP (for break point, the same way you would in GSBug) to get access to the break point window. Next, select the line (or half the line since you can only see 40 characters) where you want the break point to be placed. The line number for that line will then appear in the break point window so you have a vague indication of where the break point is. The ORCA/Debugger offers a break point feature which Splat! doesn't—the ability to only trigger a break after the break point has been passed a certain number of times. While this *is* a useful feature, I rarely use it. To set a break point in Splat!, simply highlight the line you want to break on in the source code display window and type "B". A diamond will then appear next to that line so you can see exactly which line will be broken on.

Now let's set an auto-go range in the Splat! debugger. Highlight the line where you want the auto-go range to start, press "A" to start the range, then move to the end of the range and press the return key. A vertical line will then appear next to all of the lines in the range you selected so you can see exactly which lines will be executed at full speed. To set an auto-go range with the ORCA/Debugger, you type MP at the command line to get access to the auto-go window. (The MP stands for Memory Protection—GSBug has a memory protection window, and its purpose is similar to an auto-go range, but with a high level language, the I find that the memory protection wording is confusing and misleading.) Next select the range of lines where you want the auto-go to be set for. Again, like the break point display, the line numbers are shown in the auto-go window giving you a vague idea of where the auto-go range is.

Gripes

OK, now I'll tell you everything I didn't like about both products that I haven't said before. First I'll comment on the ORCA/Debugger. First off, typing characters in the ORCA/Debugger is not recognized very consistently. More than



SPLAT!

likely, the debugger will miss one or two of your keystrokes. This is very annoying. Next, sometimes the debugger refuses to do its job and just hangs the system (but only for programs that have debugging code in them—it won't crash things you aren't trying to debug). For Splat!, navigating the dialogs sometimes requires guesswork as to what key is mapped to which text button. For simple dialogs with OK and Cancel buttons, you're pretty safe assuming that OK is return and Cancel is escape. For dialogs with more than those buttons, you're in the dark. It'd be nice if some indication of the valid keys were present. Also, when viewing variables, you can only see them in one format. By this, I mean that if you're viewing an integer, you can only see it as a signed decimal integer. Sometimes it's more useful to see it as a hexadecimal word value, or maybe even as characters. Splat! allows the use of GSBug-style templates so that user-defined structures can be made up and the applied to variables in order to view memory in a structured manner, so I thought that I could possibly use templates to view my integer as a hexadecimal word. I loaded in the set of templates that comes with Splat! and tried to assign the word template to my variable. Nothing happened. I tried assigning a lot of other templates. Still, no results. I just couldn't get templates to work. With a little help from Procyon, I figured out what I was doing wrong: to see variables using templates, you have to "expand" the variable like you would if you were viewing the contents of a record or structure. And even then, there's no way to see variables as unsigned hexadecimal words, the word template is actually the same as an unsigned decimal integer view. You can, however, assign the char template to see a variable as a character just fine.

For both products, one of the biggest things that annoys me are the keys they use to switch from the debugger screen to the graphics screen and back. I'm used to GSBug having T turn on the debugger screen and S turn on the super hi-res screen. With the ORCA/Debugger, the S key turns on the super hi-res screen all right, but the T key turns on the 80 column text output screen, not the debugger screen. You have to use the N key to get back to the debugger screen. With the Splat! debugger, the 0 key turns on the super hi-res screen, the 8 key turns on the 80 column text output screen, and the D key gets you back to the debugger screen. It'd be nice if they'd all agree to use the T key to get back to the debugger display at least.

Buggeroo!

I guess I've come to the point now where I have to recommend one product over another. If I were spending my money on a debugger, I think I'd rather have Splat! than the ORCA/Debugger. There are many reasons for this, not the least of which is that the Splat! interface is more visual and gives you better control over the debugging process. The ORCA/Debugger is patterned after GSBug, and while GSBug is excellent for debugging machine language, its cramped visual interface is not very well suited for debugging programs written in a high level language. ***GS+***

Buying Ad Space in GS+ Magazine

Full Page Ads
Each page should be 7 & 1/8 inches wide and 9 & 1/2 inches tall
1 page - $150
2 pages - $250

Partial Page Ads
Partial page ads can be any shape or orientation, but must be no more than 7 & 1/8 inches wide and 9 & 1/2 inches tall
1/3 page (no more than 22.5 square inches) - $60
1/2 page (no more than 33 square inches) - $90
2/3 page (no more than 45 square inches)- $120

If you want your ad to appear on the back cover or inside back cover, add $100. Call for cover reservations.
Spot color is available for an additional $15 per page. Call for details.
Save an additional 20% by placing the same size ad in more than one issue!
Prices good through July 1, 1994
Request your ad space today to guarantee these low prices!

DEADLINES:
May-June 1994 (Volume 5, Number 5) May 6, 1994
July-August 1994 (Volume 5, Number 6) July 1, 1994

All ads must be camera-ready copy. Payment must accompany ad. Make checks payable to **EGO** Systems, or call us to charge it on your credit card. For more information contact:

GS+ Magazine
P.O. Box 15366
Chattanooga, TN 37415-0366
Voice phone: (615) 843-3988

If you need to ship your ad to us using a service other than the U.S. Postal Service, please call to make arrangements.
If you wish to place an ad for a product we have not reviewed, we request that you include a review copy with your ad.

ORCA/Modula-2

By Peter Easdown

Retail price: $150
Typical mail-order price: $80

Not copy protected
Requires 1.25MB of RAM, at least one 3.5-inch drive, and System 6 or later. At least 2MB of RAM and a hard disk are *strongly* recommended.

Byte Works, Inc.
4700 Irving Blvd. NW, Suite 207
Albuquerque, NM 87114
(505) 898-8183

Reviewed by Steven W. Disbrow

ORCA/Modula-2 is an Apple IIGS implementation of the Modula-2 programming language that was designed by Niklaus Wirth. If that name sounds familiar, it's because Mr. Wirth is also the fellow that gave us the Pascal programming language. In fact, Modula-2 is kind of a descendant of Pascal, and the two languages are very similar. So, if you know how to program in Pascal, you will feel right at home writing a program in Modula-2. But, this is not to say that there are not some major differences between Pascal and Modula-2. (FYI: There *was* a Modula-1 [but it was simply called Modula] and there is a Modula-3, but they aren't quite as popular as Modula-2.)

The first difference is Modula-2's focus on the concept of expressing programs in distinct, self-contained *modules* (which, as you can imagine, is where they got the name "Modula"). Basically, the goal is to break a program down into several different modules, each of which imports from the others only what it needs to perform its function. If this sounds a bit like the concept of Pascal Units, it is, but it goes beyond Units. (Actually, Units were an *addition* to Pascal, while the module concept was a part of Modula-2 from the start.) You see, each module can itself be composed of two different modules a definition module and an implementation module (this is similar to the interface and implementation parts of a Pascal Unit). The definition module lists all of the constants, types, variables, and procedures that are available to other modules, and the implementation module contains the actual Modula-2 source code that implements those procedures. An implementation module can also contain constants, types, variables and procedures that are *not* listed in the definition module, but those items will be local to the implementation module and therefore hidden from all other modules. And, since the definition and implementation parts are in two physically separate source code files, if you only change the implementation part, that's all you have to recompile!

The second major difference between Modula-2 and Pascal is that Modula-2 has built-in support for *multiprogramming*. Multiprogramming is where two or more processes execute concurrently. However, I have not had the chance to work very much with this aspect of ORCA/Modula-2, so I really can't say that much about it, except for the fact that, "it's in there."

The final difference is that Modula-2 is a *very* strongly typed programming language. What's that you say? "Pascal is a strongly typed language too!" Hah!

Straight Jacket or Padded Cell?

If you thought Pascal was a strongly typed, highly structured language, forget about it. Modula-2 is so strict, it makes Pascal look like C with prototyped headers!

But, Modula-2 isn't strict just because it has tough type checking rules, it has two other traits that make it a tough language. First of all, it's a case sensitive language, and *all* of the language keywords *must* be in UPPERCASE. Think about that for a second—a language with the tough type checking of Pascal and the case sensitivity of C. Think of all the tiny, nit-picky mistakes that you will make. (Oops. Now you have soiled your trousers. Go change. And I'll wait for you to come back.)

Finally, there is this nasty (or nifty, depending on your point of view) mechanism called the `IMPORT` statement, which you *must* use to access the contents of other modules. It works like this: If you want to use, say, the `NewWindow2` tool call in your module, you *must* code the following statement at the top of the module:

```
FROM WindowManager IMPORT
NewWindow2;
```

Want to use one measly constant? You have to import that too:

```
FROM aModule IMPORT
someConstant;
```

In other words, you have to import *every single constant, type, variable, and procedure* that you want to use from another module!

However, there is a positive side to all of these seemingly restrictive things:

Type Checking: Entire books have been written about the virtues of Pascal and its strict type checking, so I can't really add anything new here. However, I will say that Modula-2 offers an extremely intelligent way to get around the strict type checking, which I'll talk about a bit later.

Case Sensitivity: All by itself, as it is implemented in C, case sensitivity is a royal pain in the rear. But, with the additional restriction that all Modula-2 keywords must be UPPERCASE, your source code becomes a *lot* more readable. This pays off in a big way when you are tracking down logic errors and code structure problems.

`IMPORT`: Without a doubt, using the `IMPORT` statement is a pain. But, it too has a big payoff in the fact that it *forces* you to *think* about what you need to use in order to get the job done. In addition, it also gives you a great reference, at the very top of your source code, as to just exactly what you are using in a particular module.

And, I should also mention that there is a way to get around the worst aspect of the `IMPORT` statement that actually leads to another neat feature of Modula-2. If you want to use *everything* in a particular module, you can just say:

```
IMPORT myModule;
```

The only catch is that to actually *use* anything from that module, you have to fully qualify the name of whatever it is you want to use. For example, to assign a zero to the variable `aNumber` inside `myModule`, you would have to code:

```
myModule_aNumber: = 0;
```

Now, if that looks like a pain, consider that this scheme allows you to access constants, types, variables and procedures with the same name from different modules! So, if you don't like the way `NewWindow2` works, you can write your own, and code something like:

```
myModule_NewWindow2(...);
```

And you can do this without giving up access to the standard `NewWindow2` routine!

Personally, I must admit that at first, I was *very* put off by all of the safeguards and restrictions that are built into Modula-2. However, after working with Modula-2 on a real project, I've actually come to view it as a "padded cell" that's there for my protection, rather than a "straight

jacket" that's there just to make my life miserable.

Other Language Differences

If you have never programmed before, and Modula-2 was your first language, the things I'm about to discuss would seem completely natural to you. However, if like myself, you've programmed in several different languages, there are a few things about Modula-2 that will take some getting used to . . . .

• The "_" character is not a valid character for use in naming things. That's because it is used (as shown above) to connect procedures, constants, etc. to the name of the module they are contained in. (Am I nit-picking here? Perhaps, but the first Modula-2 procedure I tried to write was called "Hello_Finder" and the compiler did not like it at all!)

• For a language that is descended from Pascal, it was quite a surprise that there is no support for "standard" Pascal strings (i.e. a length byte followed by the text of the string). Modula-2 strings are basically C-strings (text followed by a terminating null). Fortunately, ORCA/Modula-2 contains a full set of string conversion routines that you can easily use to switch between string types.

• There is no FUNCTION keyword as there is in Pascal. In Modula-2, functions are simply PROCEDUREs that just happen to return values.

• Unlike Pascal, Modula-2 allows you to program "on the metal" via the INLINE procedure. As you might guess from the name, this procedure allows you to enter machine code right in the midst of your Modula-2 code. This is great for making pit-stops in GSBug and hard coding breakpoints for use with Splat! or the ORCA/Debugger.

• Bit manipulations are accomplished via the use of set arithmetic. In other words, there is no bitwise AND operator or any bitwise operators at all! This one threw me for a loop for a little while because there really isn't any discussion of this in the ORCA/Modula-2 manual. And, to make matters worse, ORCA/Modula-2 uses sets to implement a *lot* of the flag bits fields that are used by GS/OS and the Toolbox. However, after I figured it out (by looking at the sample source code that comes with ORCA/Modula-2), I became quite comfortable with using sets to twiddle my bits.

• Modula-2 can use the same type casting mechanism as most versions of Pascal, but you shouldn't use it. Why not? Because Modula-2 also has its own unique type casting mechanism, the VAL procedure, that makes code maintenance a breeze. For example, consider the following Pascal-style type casting:

```
myWordPtr = wordPtr(
myLongPtr);
myCharPtr = charPtr(
myWordPtr);
```

Next, consider the equivalent statements that use the VAL procedure:

```
myWordPtr = VAL( wordPtr,
myLongPtr);
myCharPtr = VAL( charPtr,
myWordPtr);
```

Now, let's suppose that you need to find and change every single type cast in your program. Let's also suppose that there are *hundreds* of these little blighters in your code. Which would you rather be faced with; Separate find and replace operations for "wordPtr(" and "chrPtr(" or a single find and replace for "VAL("?

Now that I've given you a brief overview of the Modula-2 language and how it differs from Pascal, let's look at the ORCA/Modula-2 implementation in more detail. (Just in case you were wondering how I became an "expert" in Modula-2, I used ORCA/Modula-2 to write the program What Is This? which is in this issue of *GS+* Magazine. After all, the best way to learn a language is to write some real software with it.)

The Documentation

As is usual with ORCA languages, the documentation for ORCA/Modula-2 is very good. Every facet of the compiler and the ORCA environment is covered in great detail; however, there are a few minor problems. There are several typos in the manual and there is no "H" heading in the index. But these are extremely minor problems compared with the overall quality of the documentation. A more serious problem is that, as usual, the manual assumes that you already know the language. In the cases of ORCA/C and ORCA/Pascal, this was no big deal, because there are literally hundreds of books available about these languages, and the ORCA manuals for these languages each contained a list of books that you could easily get to help you learn them. However, this is not the case with ORCA/Modula-2. It mentions only one Modula-2 book, *Programming in Modula-2* by Niklaus Wirth. This book, even though it is the basis of the Modula-2 language, is not exactly a great book for learning Modula-2. Not only that, but it is hard to find. Beginning Modula-2 programmers would probably be better off getting the book *Oh My! Modula-2!* by Doug Cooper (the co-author of *Oh! Pascal!*). Or if you are already familiar with the Pascal language, you might want to check out the book *Modula-2 for Pascal Programmers* by Richard Gleaves. (Buyer beware, I have not actually read these books; I have only glanced at them in the computer book store. However, I am *very* familiar with Mr. Wirth's book, and I think that anything would be better than it!)

All this is not to say that the Modula-2 documentation in the ORCA manual is useless. In particular, you should read chapters 21, 22, and 23, as they contain some very useful information about the ORCA implementation of Modula-2.

The Environment

ORCA/Modula-2 is an ORCA programming language intended for use with either the ORCA shell or the Apple Programmers Workshop (APW). (Note that APW is actually an old version of the ORCA shell environment. In most contexts, the two terms are interchangeable [i.e. "APW" and "ORCA shell"]. However, APW is *really* old, and I can't recommend that anyone actually *use* it over the ORCA shell.) The ORCA shell is a text-based development environment that borrows many of its concepts from UNIX. ORCA/Modula-2 also works with the Prizm desktop-based development environment that allows you to build your programs with access to your New Desk Accessories, and all of the other neat stuff that makes your IIGS a IIGS. ORCA/Modula-2 includes both the shell and Prizm, so you can try out both environments and decide which one you prefer. ORCA/Modula-2 also comes with a whole slew of programming utilities that you can use with either the shell or from inside Prizm.

Prizm also has an extremely cool editor that lets you set and clear breakpoints (for the debugger), and edit your files in a "split window" view (i.e. you can edit two different parts of a file at the same time, in the same window).

In the past, the Prizm environment has been (and I'm being nice here) a little flakey. However, this latest version seems to be very solid. Unfortunately, in my experience, actually using Prizm to work on a program has been more trouble than it's worth. The whole "compile, link, test" cycle becomes overly complicated and time consuming because Prizm tends to just get in the way. However, Prizm is probably what most beginners will want to use, simply

because most of the tutorials are presented in the Prizm environment.

The Compiler

Without a doubt, this is the most robust and bug-free ORCA compiler that I have used. That's not to say that there aren't any bugs; however, I have only found two so far, which I'll mention shortly.

In addition to being extremely stable, the ORCA/Modula-2 compiler is very fast (thanks in part, I suppose, to the fact that the use of the IMPORT statement eliminates the time that is spent in other ORCA languages importing unused constants, procedures, etc.). On top of that, the ORCA/Modula-2 compiler also gives very good feedback during the compilation process. For example, in the other ORCA compilers, when an error is flagged, the exact location of the token that caused the error sometimes seems to be off by a token or two. Modula-2 gets it right every time. Also, when the compilation is complete, the ORCA/Modula-2 compiler tells you, in bytes, just how much code was generated, and how many data bytes were generated. This is a nice touch that I hope makes its way into the other ORCA compilers.

And, since it is an ORCA compiler, it's fairly simple to mix your Modula-2 code with code written in any of the other ORCA languages. This can be extremely useful if, say, you can't stand the use of sets for bit manipulation—just write your own bit twiddling functions in ORCA/C!

So, what's not to like? Well, there are several things that I've come to expect in an ORCA compiler that are simply *missing* in ORCA/Modula-2.

First of all, there are a few compiler directives that I really miss: optimize, toolparms, and debug. The lack of the optimize directive leads me to wonder if ORCA/Modula-2 does any optimizations at all! However, the code that it produces does seem fairly compact and fast, so it may be that an optimize directive simply wasn't needed. The missing toolparms directive also caused me a few moments concern; after all, What Is This? is a Finder extension, so I thought it would require something like a toolparms directive to work correctly. However, I found that Modula-2's RTL directive worked just fine for this purpose. The biggest disappointment however, was the lack of a debug directive. Without this directive, you are forced to use the "+d" flag in the command line of your compile command to turn debugging code on for the entire module. This is very annoying when all you want to debug is a single procedure. I suppose however, that if you split your program into enough modules, this would be only a minor annoyance.

Speaking of debugging, another disappointment is that ORCA/Modula-2 simply does not generate the debugging information needed to look at records and other complex structures in the ORCA/Debugger and Splat!. However, the Byte Works tells me that they hope to include this feature in an updated version of ORCA/Modula-2.

The Two Bugs

While the above items are simply annoyances, there are two bugs in the compiler that you should be aware of:

First of all, if, during the compilation process, the compiler is paused and you press Command-period to stop the compilation, the keystroke will be ignored and the compiler will continue on with the compilation. This is *extremely* frustrating, especially if the compiler has just flagged an error and you want to stop compiling and go to the editor to fix the problem. However, if the compiler is not paused, Command-period works just fine.

Second, if you are coding a CASE statement, make sure that you include an ELSE clause in the case statement. If you don't, and there is a instance that is not handled by your CASE statement, your program will go off into the weeds somewhere, usually forcing you to reboot. For example, if the value of x is "3", the following will cause trouble:

```
CASE x OF
   1 : SysBeep();
|  2 : FlashMenuBar();
END;
```

However, if you code:

```
CASE x OF
   1 : SysBeep();
|  2 : FlashMenuBar();
   ELSE ;
END;
```

everything will work fine.

Other Problems

The only other problems with ORCA/Modula-2 are, once again, problems of omission. For example, there is no interface file for communicating with the Finder via IPC. So, I had to make a partially complete one of my own (this file is on your *GS+* Disk with the source code for What Is This?). Also, there is no interface file containing resource type definitions, and some of the constants used by the TextEdit tool set are missing. However, if you have ORCA/Pascal, just about all of these missing things can be plundered from its interface files with only minimal changes, so these omissions aren't a fatal flaw in ORCA/Modula-2.

Conclusion

If you already have ORCA/Pascal and/or ORCA/C, you are probably asking yourself, "Do I need ORCA/Modula-2?" Well, if you are happy with ORCA/Pascal or ORCA/C, and you aren't interested multiprogramming, probably not. However, ORCA/Modula-2 is a *very* solid compiler, and a good, usable language for IIGS programming. And, if you have reached the end of your rope fighting with the ORCA/Pascal and ORCA/C compilers, I would say "Yes," ORCA/Modula-2 is a worthwhile purchase simply because it is the most bug-free ORCA language I've used. And, if you are interested in learning more about multiprogramming concepts, ORCA/Modula-2 is just about the only game in town. *GS+*

Professor Gumby's Top Ten Rumors of All Time
Number 4 - From *GS+* V2.N2
November-December 1990

Sequel Fever!

According to a spokesman for Electronic Arts, if sales for The Immortal, Keef the Thief, and Pipe Dream (which EA distributes) are good, we should soon see some sequels to these programs. The sequel for The Immortal will be called, "The Guy Who Falls In Holes and Dies a Lot." Seems like a fairly self-explanatory plot. The sequel to Keef the Thief will be called "Flo the . . ." Hold on! That can't be right! Must be a mistake on this note that Diz gave me. To continue: The sequel for Pipe Dream will be called, um . . . we can't print that! Well, anyway, in this game you have to connect these, er, uh . . . hey! Who's sick idea was this anyway?

LASERbeam v1.1

By Josef W. Wankerl

When I first wrote LASERbeam, I didn't think that too many people would actually have a use for it. To my surprise, we've received quite a bit of positive feedback from people who have used it. It's been a year since LASERbeam was first introduced, and now it's time to add a feature that I wanted LASERbeam to have when it was first conceived: PostScript® Type 1 font downloading.

Note that the name LASERbeam implies laser printer support only. This is not true. While we only have a LaserWriter IINT here at the office to test LASERbeam with, you should be able to send PostScript files to any printer device that you can connect to your IIGS. You can even send PostScript files to devices which do not support PostScript, such as an ImageWriter II. Of course, if you do this, you probably won't get very pretty results. (If you do send the text directly to an ImageWriter II, be sure that you have the printer set to force a line feed after carriage returns or all the lines will overwrite themselves.)

What's New

If you are a prior user of LASERbeam, you'll notice that not that much is actually new in the user interface department. There's one more item in the Download menu to let you download Type 1 fonts, and there is a completely new Special menu. The Special menu just lets you quickly download some common useful PostScript programs to the printer, so not much work went into that area.

Installing LASERbeam

To install LASERbeam, refer to "How to Use Your *GS+* Disk" in this issue. LASERbeam can only be used with System 6.0.1 or later. If you don't wish to install the LASERbeam program, you can just run it directly from your backup *GS+* Disk.

Using LASERbeam

LASERbeam is more than just a utility to send PostScript files and fonts to your printer, it is actually a PostScript program authoring system! To demonstrate this, let's create a new PostScript program and send it to the printer. First, select the New PostScript® menu item from the File menu. You'll be presented with a new, empty PostScript document window. Now you can type in a PostScript program. For a small sample program, see Figure 1. Once you have your PostScript program written, you can tell the current printer to interpret it by using the Front Window menu item from the Download menu. If you're proficient in PostScript, you'll probably enjoy how you can work on your program and send it to the printer (using the Front Window menu item from the Download menu) without leaving LASERbeam. There are some sample PostScript programs on your *GS+* Disk for you to play with if you don't know PostScript, but you would like to see how LASERbeam works. (My favorite is called **GSPlus**.) You can use the Open PostScript® File menu item from the File menu to load the **GSPlus** file into a PostScript document window. You can then use the Front Window menu item from the Download menu to send it to the printer. If all you want to do is send the file to the printer without actually loading it into a document window first, you can use the PostScript® File menu item from the Download menu.

Downloading Fonts

The real gem included in this new version of LASERbeam is the ability to download Type 1 fonts to your printer. A Type 1 font is actually a special PostScript program which defines how the font is generated. There's no real trick to downloading a Type 1 font with LASERbeam—all you do is choose the PostScript® Font menu item from the Download menu, choose the font you want to download, and then sit back and wait. All the magic takes place behind the scenes. Before you go off and download every single font in the world, you should probably know a bit about how your IIGS uses the LaserWriter driver to talk to the printer and find downloaded fonts.

When you print a document, the LaserWriter driver queries the printer to find out what fonts are available. The list of fonts it gets back is by name only. This means that if you use a font called "Gonzo" in your document, in order for it to print correctly, a font called Gonzo must be present in the printer. (Note that font names *are* case sensitive, so "Gonzo" and "gonzo" are two different fonts.) So far so good, right? Well the tricky part comes when you add styles to your text. If your document has the font with the bold style applied to it, the LaserWriter driver looks for a font called Gonzo-Bold. If that font isn't found, it looks for just plain old Gonzo and tells the printer to apply the bold style, but it probably won't look as good as if you had actually downloaded the actual bold font. The same rule applies for the italic style—the LaserWriter driver looks for a font called Gonzo-Italic. If you apply both the bold and italic styles, the LaserWriter driver looks for a font called, surprisingly enough, Gonzo-BoldItalic. Now, to throw you a curve, Type 1 fonts are not always (actually almost never) named according to the above conventions. You're pretty safe when you download a plain font, but when you download a styled font, chances are its name is Gonzo-BookOblique, not Gonzo-Italic. This can cause a major headache when you try to use that font and the LaserWriter driver can't find it. LASERbeam goes through a *lot* of trouble to keep all of this hidden from you, though. When you download a font, it actually looks inside the font program and will change "Gonzo-BookOblique" to "Gonzo-Italic" so the LaserWriter driver will be able to find it. For the last word on the LaserWriter driver font naming rules, you should read IIGS Technical Note #67 which is included on your *GS+* Disk in the **Documentation** folder. (Please note that Technical Note #67 needs to be revised now as it contains some incorrect

Figure 1

```
%!PS
%
% Print a box, centered on the page
%

newpath
          270 360 moveto
          0 72 rlineto
          72 0 rlineto
          0 -72 rlineto
          closepath
          3 setlinewidth
          stroke showpage
```

information. Specifically, it states, "Currently there is no means to download a PostScript font with an Apple IIGS." This is, of course, in error.)

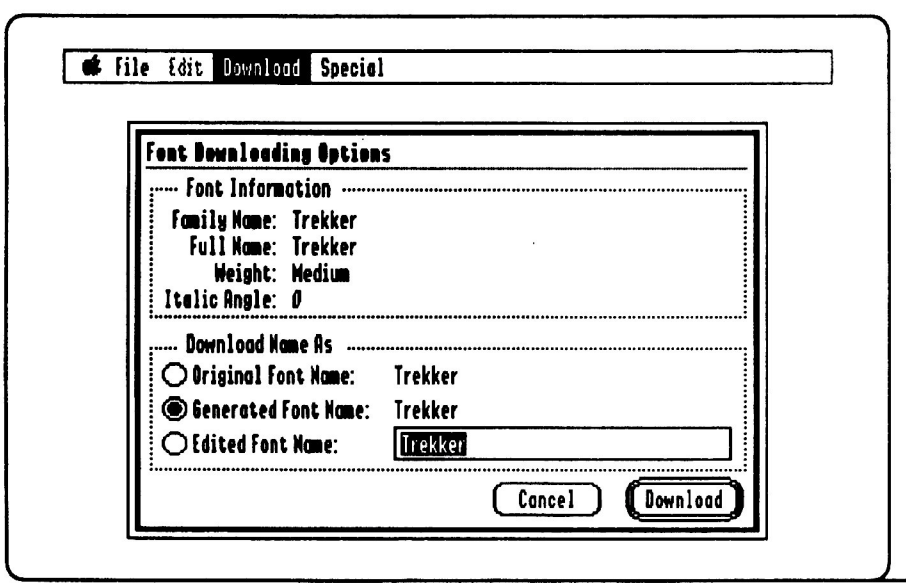
So why am I telling you all this if LASERbeam does name conversions automatically? Well, if you hold down the option key when you choose the PostScript® Font menu item from the Download menu, a dialog box will appear (see screen shot) letting you change the name of the font yourself. The top half of the dialog is informational only—it shows you the font attributes that LASERbeam found inside the font program. The four attributes are the font's family name, the font's full name (which is also the name known to the printer), the font's weight (how thick or bold the font's lines are—common weights are "Medium" meaning plain and "Bold" meaning bold), and the font's italic angle (with zero being no degree of italicization). The lower half of the dialog is where you can tell LASERbeam what to do with the font name. The three radio buttons give you a lot of control over the name of the font. If you choose the Original Font Name radio button, the font will be downloaded without any changes. If you choose the Generated Font Name radio button, the font will be downloaded with the LaserWriter driver style font name that LASERbeam generated. If you choose the Edited Font Name radio button, the font will be downloaded with the name that you type into the LineEdit control. Using the Edited Font Name radio button can be fun—I actually downloaded a font to the printer and instead of calling it by its real name, I titled it Venice. I then printed a document which used the Venice font (which every IIGS user should have—it comes on your System disks) and the printout used the downloaded font instead of Venice. It's kind of a sneaky trick, and I don't recommend you do it for serious work, but it was a fun experiment to try. I also tried naming a font Shaston, but it seems that the LaserWriter driver always special cases Shaston and prints using Helvetica instead.

Using Fonts

OK, so LASERbeam lets you download PostScript Type 1 fonts to your printer. How do you actually use them? Well, you create a document using a font with the same name. Since the IIGS doesn't know about Type 1 fonts, you're in the dark for actually doing that (unless you're lucky and have a bitmapped font with the same name, which is unlikely). If you have Pointless, chances are you can find the same font in a TrueType representation. You can then use the TrueType font to create your document on your IIGS and then rely on the Type 1 font downloaded to your printer to handle the printing. If you're going to go out and find some good fonts to work with, a good thing to do is to get a collection of fonts that include fonts both in Type 1 and TrueType formats. A good example of this is the KeyFonts Pro collection, reviewed in *GS+* V4.N5.

Debugging Mode

In order to get this version of LASERbeam working correctly, I needed to see what LASERbeam was doing to the font before it was downloaded to the printer. To prevent the headache of downloading a bunch of broken fonts to the printer, I put some code in which, instead of sending code to the printer, sends it to a result window instead. This let me check the nifty font name manipulations without cluttering up the printer. It also let me check what the items under the Special menu were sending to the printer. In case you're curious, you can turn debug mode on by holding down the option key and choosing the About LASERbeam item from the Apple menu. You turn debug mode off by choosing the menu item with the option key down again. Note that when you download something with debug mode turned on, the result window will contain a valid PostScript program which you can save and then later download.



The Menu Bar

The LASERbeam menu bar is a fairly typical menu bar, so let's take a few moments to go over each menu, one at a time.

The Apple Menu

The first item in the Apple menu is the About LASERbeam item. Selecting this item presents you with a window telling you the version of LASERbeam as well as some information about how much memory you have available. To get rid of this window, click the mouse in its close box or select the Close item from the File menu. If you select the menu item while holding down the option key, LASERbeam's internal debugging mode is toggled. For more information on the debugging mode, see the "Debugging Mode" section above.

Below the About item will be a list of your installed new desk accessories. To use one, simply pick it from the menu and then use it as you normally would.

The File Menu

The first item in the File menu is the New PostScript® item. This item simply creates a new PostScript document window so you can type in a new PostScript program.

The second item in the File menu is the Open PostScript® File item. This item presents you with a Standard File dialog letting you choose a PostScript program to load. When you choose a PostScript program file, it is loaded into a PostScript document window.

The third item in the File menu, the Close item, allows you to close the topmost window. If the window is a PostScript document window that has not yet been saved, you will be asked if you want to save the file before closing it. If you answer Save, the document will be saved as if you had picked the Save menu item. If the document has never been saved before, you will be presented with a Standard File dialog box that will allow you to save the document under a new name (as described under the Save As item below). If you answer Don't Save,

the PostScript document window will be closed and any changes will be lost. If you answer Cancel, the operation will be cancelled and the PostScript document window will not be closed.

The next item, the Save item, allows you to save the contents of the current PostScript document window. If the PostScript file has not already been saved, you will be presented with a Standard File dialog box that will allow you to save the PostScript file under a new name, just as if you had selected the Save As item (described below). If the PostScript document has already been saved, it will be saved to the same file name as before.

The next item in the File menu, the Save As item, allows you to save the contents of the current PostScript document window to a new file name. When you pick this item, you will be presented with a Standard File dialog box that will allow you to specify a new file name to save the PostScript document under.

PostScript documents created with the New PostScript® menu item are always saved as PostScript document files. PostScript documents that were opened using the Open PostScript® File menu item are saved as either PostScript document files or plain text files depending on the original format of the file. You can load these files into any text editor (like EGOed lite) that can read text or source code files (since a PostScript document file is actually a type of source code file).

The next item in the File menu, the Page Setup item, lets you change the page setup information that will be used to print your PostScript code. For best result, I recommend that LaserWriter owners always set the Vertical Sizing option to Condensed and that ImageWriter owners set the Height option to Vertical Condense. If you don't, your printout will look "stretched out" vertically. (Note that this bit of advice can be applied to all IIGS desktop programs, not just LASERbeam!)

The sixth item in the File menu, the Print item, allows you to print your PostScript code to your currently selected printer. If you select the Print item and you have not yet specified a page setup, the Page Setup dialog will appear, just as if you had selected the Page Setup menu item. After you have specified a page setup, you will be presented with the Print dialog that will allow you to specify the number of copies you want to print and the pages you want printed. Note that the Print menu item actually prints a copy of the current PostScript document window, it doesn't cause the PostScript program to be executed on the printer—to do that, use the Front Window menu item from the Download menu.

The last item in the File menu, the Quit item, lets you quit from LASERbeam and return to the previous application. If you have any open PostScript document windows that have not been saved, you will be asked if you want to save them before quitting.

The Edit Menu

The Edit menu is used by LASERbeam just like it is from any program that supports the Edit menu. If the front window is a PostScript document window, you can select text in the report window using the mouse and then use the Cut, Copy, Paste, and Clear items in the Edit menu just as you would in any other application. For more information on how these editing commands work, refer to your IIGS owners manual.

The last menu item in the Edit menu, the Select All menu item, causes all of the text in the current PostScript document window to be selected.

The Download Menu

The first item in this menu, Front Window, sends the PostScript code in the current PostScript document window to the current printer. If the current printer is a PostScript device, the PostScript code will be interpreted by the device.

The second item in this menu, PostScript® File, presents you with a Standard File dialog asking you to choose a PostScript document to send to the current printer. The PostScript code in the PostScript document you select will then be sent to the current printer. If the current printer is a PostScript device, the PostScript code will be interpreted by the device. The advantage of using the PostScript® File menu item instead of the Front Window menu item is that the PostScript® File menu item does not have to have the PostScript document open, so very large PostScript documents can be sent to the printer even if they would not normally be able to fit into memory.

The last item in this menu, PostScript® Font, presents you with a Standard File dialog asking you to choose a Type 1 PostScript font to sent to the current printer. It may take a while for Standard File to present the list of fonts for you to choose, especially if the font you want is on a CD-ROM. This is because LASERbeam is checking the file to see if it contains Type 1 font information. For ProDOS, HFS, and AppleShare volumes, this information is readily available. For any other file system, LASERbeam has to look inside each file for the PostScript font information to know whether or not to make the file available for downloading. (A lot of CD-ROMs are formatted with the HFS file system, so waiting excruciating amounts of time to get a font list shouldn't be that common.) When you choose a font, its name is altered according to the LaserWriter driver naming rules and downloaded to the current printer. If you hold the option key down when you choose the PostScript® Font menu item, a dialog will appear letting you edit the name of the font. For a complete discussion on the font name editing dialog, see the "Downloading Fonts" section above.

The Special Menu

When you choose an item from the Special menu, a program is downloaded to the current printer. Any output from the printer is *not* displayed in a result window. (See the "PostScript Results" section below for a description of a result window.) If you want to see the results, you must download the program using the menu items from the Download menu. The equivalent programs are contained on your *GS+* Disk. Note that these special programs have only been tested with Apple LaserWriter printers. They'll probably work on other printers, but I won't guarantee it.

The first item in this menu, the Reset Printer item, presents you with a dialog asking if you really want to reset the printer. Resetting the printer causes all the downloaded fonts in the printer to disappear. If you choose to reset the printer, the reset program is sent to the printer.

The second item in this menu, the Turn Start Page On item, sends a program to the printer telling it to enable the printing of the test page when you turn on your printer.

The third item in this menu, the Turn Start Page Off item, sends a program to the printer telling it to disable the printing of the test page when you turn your printer on.

The last item in this menu, the Print Font List item, sends a program to the printer telling it to print a sample of all the fonts currently in the printer's memory. You can use this menu item to make sure that a font was actually downloaded to the printer, plus it makes a handy reference chart.

PostScript Results

Some PostScript programs produce results, not on paper, but back to the computer that sent the program. An example of one such program is the **TestPageOff** program. The program disables the test page printing on a LaserWriter when it is first turned on, saving countless trees. The result string passed back to LASERbeam says that the permanent state of the LaserWriter has been changed. Any results from a PostScript program will be shown in a separate "PostScript Results" window. You can save the window contents in a standard text file if you wish. If a PostScript program does not generate any textual results, the result window will not be displayed. (Note that you can use another PostScript program on your *GS+* Disk, **TestPageOn**, to turn the test page back on.)

LASERbeam Tricks

OK, so LASERbeam is another gee-whiz program with no practical applications, right? HA HA! Fooled you yet again!

LASERbeam is an excellent environment for teaching yourself PostScript. You can write and test your own PostScript programs! It's really easy to follow along in the PostScript cookbook/tutorial using LASERbeam, too.

You can use LASERbeam to blast text to a directly connected printer. You'll receive a warning saying that sending PostScript to a non-PostScript printer will be unpredictable, but if you know what you are doing, you can quickly send a text file to an ImageWriter II printer, or any other printer that accepts text directly. (Don't forget to make sure that line feeds are generated after carriage returns.)

But the most useful function of LASERbeam is its ability to print PostScript document files that were created by pressing Command-F from the Print Manager's print dialog. If you are printing to a laser printer, you can press Command-F to create a PostScript document file instead of having the file print. To do this, when you're at the print job dialog, just hold down Command-F and then click on the Print button. You should see a message saying that a PostScript file is being created. The file will be saved in the ***:System:Drivers** folder and it will be called **PostScript.GS*XX*** where XX is a number. If you don't have a laser printer connected to your IIGS, but you have the LaserWriter printer driver installed, you can use this trick to save the PostScript file and then take that file to a IIGS connected to a laser printer and use LASERbeam to actually print the file. This way you don't have to have the original application and document to get your output, you just need the PostScript document, a PostScript device, and LASERbeam.

Inside LASERbeam

If you're not a programmer, you can skip the rest of this article. If you are a programmer and are interested in what is actually going on inside LASERbeam, especially in the realm of downloading Type 1 fonts, then by all means read on. I mentioned previously, "there's no real trick to downloading a Type 1 font with LASERbeam," but that's only from a user's standpoint. There are quite a few interesting things to pick at inside the LASERbeam source code.

First off, a Type 1 font is defined by a group of Macintosh "POST" resources. (OK, so you can find Type 1 fonts in other formats, but LASERbeam only knows about the ones in Mac resources.) The font definition starts in resource type "POST" with ID 501 and continues with consecutive IDs. The first two bytes in the resource define what to do with the rest of the data contained in the resource. (This information is found in the *LaserWriter Reference* book.) The first byte is a value between 0 and 5. The second byte will always be zero. If the first byte is 0, the rest of the resource is considered a comment and is not sent to the printer. If the first byte is 1, the rest of the resource is plain text which should be sent to the printer unchanged. If the first byte is 2, the rest of the resource is binary data. Each binary data byte should first be transformed to its ASCII representation and then sent to the printer. For example, if the first four bytes of the resource were $FE, $ED, $F0, and $0D then the ASCII string "FEEDF00D" should be sent to the printer. If the first byte is 3, an AppleTalk end-of-file should be sent, followed by the rest of the resource data. (LASERbeam cannot send an AppleTalk end-of-file, so it just pretends like the first byte was 1.) If the first byte is 4, the data fork of the file is sent to the printer. If the first byte is 5, the rest of the resource is considered a comment and is not sent to the printer. Also, a 5 indicates that the complete font definition has been sent to the printer, so no more information should be sent. So, in order to download a Type 1 font, you simply open the resource fork, start with "POST" resource 501, interpret the code bytes, send the data, then go on to the next resource ID until you find a code byte of 5. (Note that all of the resource manipulation is done by the Miscellaneous Library's new MacResource section, and *not* the IIGS Resource Manager.)

LASERbeam has to go through a lot more trouble, though—namely, changing the font name to comply with the LaserWriter driver naming rules outlined in IIGS Technical Note #67. The routines in LASERbeam to perform this amazing feat are `GetNextToken`, `FindFontInfo`, and `FixFontName`. The first thing LASERbeam does is to call the `FindFontInfo` routine. The `FindFontInfo` routine searches through the font for the `/FamilyName`, `/FullName`, `/Weight`, and `/ItalicAngle` identifiers. To get an identifier, the `GetNextToken` routine is called. Once a font definition identifier is found, the next token is retrieved and saved. After all four identifiers have been found, the `FixFontName` routine is called to change all font name references to a name compatible with the LaserWriter driver naming rules. The `FixFontName` routine searches through the font for all occurrences of the name associated with the `/FullName` identifier (except for the one after the `/FamilyName` identifier) and changes it.

If you can handle the above, the rest of the LASERbeam program should be fairly straightforward.

P.S.

I hope you can find some additional uses for LASERbeam than I have thought up, since LASERbeam is a fairly versatile program. If you have any problems with LASERbeam, please be sure to fill out and send in a Problem Form so that I can fix them. *GS+*

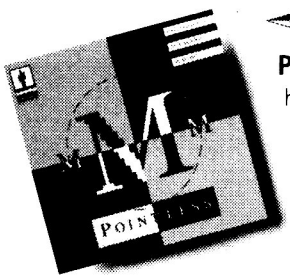
Professor Gumby's Top Ten Rumors of All Time
Number 3 - From *GS+* V1.N6, July-August 1990

Our Lawyers Made Us Say This

Just about every single word in this magazine is somebody's trademark. You can't swing a dead cat these days without getting sued (or yelled at by animal rights activists), so we just wanted to let you know that we don't make any claim to these names, we are just using them to get your attention. Please don't sue us. However, we *do* own the rights to the names "EGOed" and "Transfusion," and we will sue the pants off of any big-headed editor or hospital that we catch using them. Special site licensing is available for hospitals and anybody that publishes anything that the Bush administration might object to.

The Best of the West

Since 1990, WestCode Software has created the best software you can get for your Apple II computer. And, now for a limited time, we're offering you the best prices ever on our highly-acclaimed products.

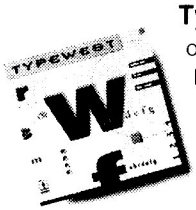


Fantastic looking text on your printer.

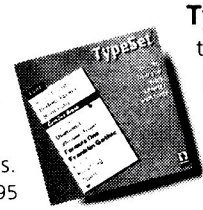
Pointless 2.0 All your documents will look better than ever before, without unsightly hard-to-read jagged text. Pointless brings TrueType font technology to your Apple IIGS, providing impressive laser-sharp text on your ImageWriter, StyleWriter, DeskJet, LaserJet, and other printers. Plus, text on the screen will appear smooth at even the largest sizes.
For the Apple IIGS, System 5 or later. Reg. $69.95
Sale $39.95

The quick brown fox
The quick brown fox

ImageWriter II printing before and after Pointless



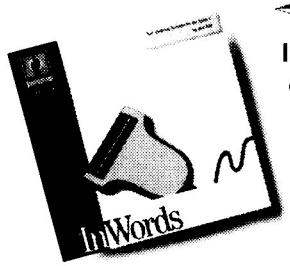
TypeWest An outstanding collection of 40 TrueType fonts for use with Pointless. These professionally-designed typefaces are perfect for home, school, or work. Don't waste your time and money on lesser quality fonts from other sources.
For the Apple IIGS with Pointless. Reg. $49.95
Sale $29.95



TypeSet The perfect companion for Pointless that helps you manage your TrueType fonts. It provides a What-You-See-Is-What-You-Get Font menu that displays your fonts in their own typefaces, font sets to work with custom groups of fonts, and extensive font reports.
For the Apple IIGS with Pointless. Reg. $49.95
Sale $29.95

Stretch your disk space.

HardPressed 1.0 If you're always searching for more disk space, then you need HardPressed — the best disk compression software for your Apple IIGS. It virtually doubles the space on any type of disk. Your 40 MB hard drive will be able to store 80 MB, or a 2 MB RAM disk can store 4 MB! HardPressed works behind the scenes, transparently compressing your files as you save them. When you open a compressed file or launch a compressed application, HardPressed automatically expands it for you.
For the Apple IIGS, System 6 or later. Reg. $69.95. AutoArk trade-up $24.95
Sale $39.95



It's like typing 500 words per minute.

InWords 1.1 Optical Character Recognition (OCR) software that let's you use a Quickie or LightningScan hand-held scanner to scan pages from books, magazines, letters, etc. into text that you can load and edit in your favorite applications: AppleWorks, PublishIt, BeagleWrite, GraphicWriter, HyperStudio, and others. NEW version 1.1 adds improved recognition, speed, line handling and more! Why type? Get InWords instead!
For the Apple IIe or Apple IIGS. Reg. $129.00
Sale $49.95

Call 1-800-448-4250

Order direct from WestCode Software and save! 30-day money-back satisfaction guarantee.

WestCode Software, Inc., 15050 Avenue of Science, Suite 112, San Diego, CA 92128 · (619) 487-9200 · Fax (619) 487-9255

Miscellaneous Library

By Josef W. Wankerl

[Editor's Note: The Miscellaneous Library is not a stand-alone program! It is a programming tool that we think advanced readers of *GS+* Magazine will find very useful. It is intended for those doing advanced IIGS programming. The information provided here is an overview of what's new in the Miscellaneous Library and does not provide complete documentation for all of its calls—if you plan to use the Miscellaneous Library, read the **MiscLib.Docs** file (which is on your *GS+* Disk) for complete information!]

The Miscellaneous Library (MiscLib) is a collection of various routines I have found myself using over and over. They can be used from any language that supports linking to standard libraries, such as ORCA/C, ORCA/Pascal, and ORCA/Modula-2. For detailed assembly language stack diagrams on how to make the calls, and for a short description of the parameters, see the figures in the **Figures** file. (This file is located in the **GSP.V5.N4.SEA** self-extracting archive that is on your *GS+* Disk.)

Overview

Welcome once again to my "department" here in the magazine. It seems that I put a new routine or two in MiscLib every other magazine. This time I've outdone myself yet again. But, before I forget, I'd like to mention that there's a new interface file for those of you who want to use the MiscLib with ORCA/Modula-2. Also, I've completely reorganized the documentation. Each "chapter" is now in a separate file. (You can now open these files with EGOed without going through a complex import procedure. All I did was split the big file up into small files—hooray for bugs in TextEdit!)

Now that that's out of the way, I've added an entirely new section to MiscLib which is basically a read-only Macintosh Resource Manager. For those of you privileged enough to own a copy of the System 6 Golden Master CD-ROM, you'll know that Apple provided its own source code to read Mac resources. (Teach uses the routines to interpret MacWrite 5.0 files.) However, the source code was very ugly (at least to me it was), very limited (there's only one call you can make), and you probably have to get some kind of license to use it. When it was decided that LASERbeam could actually be upgraded to download PostScript® Type 1 fonts, I needed a way to read Mac resources, since there aren't any IIGS Type 1 font files. I opted to write my own routines instead of Apple's. (It took me only two 9 to 5 days to get all the routines written and debugged.) Before writing the routines, I had to look extensively through *Inside Macintosh: More Macintosh Toolbox* to get the format for Mac resources. After doing my research, I figured that it would be easy to write a suite of routines based loosely on the Mac Resource Manager. I'd have to write approximately the same amount of code to load one resource (like Apple's routines) as I would to make a complete set of Resource Manager-like calls. (Most of the work in loading a resource is finding out where it resides on disk, which means traversing the resource map—once you can traverse the resource map, you can do anything.) Normally I just outline the new calls here and refer you to the **MiscLib.Docs** file for complete documentation, but this time I'll give some technical information with each call so you can see exactly what's going on. Of course, you should always read the **MiscLib.Docs** and related files as your definitive reference for all MiscLib calls. (By the way, you'll notice that all the calls start with `ml`, which, of course, stands for MiscLib. This is to keep you from getting confused into thinking that the call is a IIGS, or even a Mac, Resource Manager call.)

Resource Differences

Before you dive in, you should know how to use resources (knowing either the Resource Manager from either the IIGS or the Macintosh is sufficient—the two are similar enough so that if you know one it doesn't take much to figure the other one out). Second you need to know the difference between IIGS and Macintosh resources, namely (if you only know the IIGS Resource Manager) the way that Macintosh resources are referenced. The **MiscLib.Docs** and related files have a detailed explanation of the differences, so in order to save you the trouble of extracting the proper documentation file from the self-extracting archive and searching through the text to find it, I'll go ahead and explain the differences here.

There are a couple of differences between the Macintosh Resource Manager and the IIGS Resource Manager relating to types and IDs. On both systems, resources are identified by a type and an ID. On the Mac, the type is a longword and the ID is a word—on the IIGS, the type is a word and the ID is a longword. On the Mac, the type is usually represented by four characters and the ASCII values of those characters are used to form the type longword. On the IIGS, the ID is usually represented as an unsigned hexadecimal longword while on the Mac, the ID is usually represented as a signed decimal word. So, for example, you might want to load a "STR" resource of ID -128 on the Mac. (Note that there's a space character in that type since types are *always* four characters long.) On the IIGS, you'd probably want to load a `rPString` resource (a constant defined as $8006) of ID $0032FE20. Now, since the IIGS doesn't handle four character types in the same way that the Macintosh does, it gets a little tricky in figuring out what the actual type longword value to use is. To simplify measures, you can use the `mlStringToType` and `mlTypeToString` routines to preform proper conversions.

File Based Routines

Before you can do anything else resource-wise, you have to first open the Macintosh resource fork. The `mlOpenResFile` call will open the resource fork and return a handle which contains the resource map. When you're done working with one file, you can make the `mlCoseResFile` call which will close the file and clean up. If you need to know the refNum of the open resource fork, you can make the `mlGetOpenFileRefNum` call. (The call is named according to the IIGS Resource Manager instead of the Mac Resource Manager, since there's no equivalent Mac call.) The only other file-based call is the `mlGetResFileAttrs` call, which returns the attributes of the resource file. This call really doesn't have much significance on the IIGS, but it's included to make the suite of calls complete.

Indirect Access

Once you've got a Macintosh resource fork open, you will probably want to find out what's inside. Unless you know exactly what you're looking for, indirect access is the best way to get to the resources. (With Apple's one call you have to know exactly what you're looking for or you're out completely of luck.) There are four calls which let you know what's inside a Mac resource fork: `mlCountTypes`, `mlGetIndType`, `mlCountResources`, and `mlGetIndResource`. The two counting calls return the number of types and resources that are in the resource fork. The two indirect calls return the

Figure 1
The New Miscellaneous Library Calls

MacResource Calls	Description
mlOpenResFile	Opens a Macintosh resource fork
mlCloseResFile	Closes a Macintosh resource fork
mlGetOpenFileRefNum	Returns the GS/OS refNum assigned to the open resouce fork
mlGetResFileAttrs	Returns the resource fork attributes
mlCountTypes	Returns the number of distinct resource types
mlGetIndType	Returns a type given a type index
mlCountResources	Returns the number of distinct resources given a type
mlGetIndResource	Returns an ID given a type and a resource index
mlGetResAttrs	Returns the attributes for a given resource
mlGetResourceName	Returns the name of a given resource
mlGetNamedResource	Returns the ID of a resource given a type and a resource name
mlGetResourceSizeOnDisk	Returns the size of a given resource
mlGetResource	Loads a Macintosh-style resource
mlResError	Returns the last error encountered
mlReverseWord	Reverses the order of a word value
mlReverseLong	Reverses the order of a longword value
mlStringToType	Converts a string into a type
mlTypeToString	Converts a type into a string

identifying number of the indirect type or ID.

Name Me Beautiful

The Macintosh Resource Manager also uses resource names (in a much different way internally than the IIGS Resource Manager) to identify resources. You can use the mlGetResourceName call to find out the name of a resource if you know its type and ID. (The mlGetResourceName routine doesn't have a Mac Resource Manager equivalent, but since I'm used to having the call on the IIGS, I decided to write it anyway. It was pretty simple to do, anyhow.) You can also use the mlGetNamedResource call to find out the ID of a resource if you know its type and name. (The mlGetNamedResource call was actually the most difficult routine to write.)

Mac Resources

And now, what you're really waiting for: actually working with a Macintosh resource. The mlGetResource call actually loads a resource. The handle it returns is yours to dispose of when you're through with it. It's very important to note that the resource will be in the exact same format as it was on disk. This means that if you need to get at any of the data inside the resource, you have to remember to reverse the bytes to make sense of them (for non-ASCII data only, of course). The mlGetResAttrs call returns the attributes for a resource. Finally, the mlGetResourceSizeOnDisk call returns the number of bytes the resource takes up on disk in case you'd like to know.

Support Routines

The above calls comprise every read related call you could make with the Macintosh Resource Manager, and a bit more. There, however, are a few more routines that will make your life a bit easier when dealing with Mac resource forks. The mlResError call returns the last error that was encountered when making a Macintosh resource related call. You should make the mlResError call after every other resource call to make sure no error occurred. Since the Macintosh stores data exactly backwards from the way the IIGS stores data, I've included two routines, mlReverseWord and mlReverseLong, which will take a Macintosh-style word or long value and convert it into a form that the IIGS can use. Finally, since the Mac uses strings for its resource types, the mlStringToType and mlTypeToString routines make it easy to convert strings to and from the type identifier needed by the resource calls.

Intermediate Routines

The above resource routines are not the only new ones in the Miscellaneous Library—they get help from some support routines that you can't call directly from your program. But, in the interest of science and Fig Newtons™, I'll discuss what each routine is for and what it does. First up to bat is the mlGetIndTypePtr routine, which is a kissing cousin to the mlGetIndType routine. The mlGetIndTypePtr routine returns a pointer to a type record which is contained in the resource map. Swinging second is the mlGetTypePtr routine, which when given an actual type, calls the mlGetIndTypePtr routine repetitively until a matching type is found and then returns a pointer to that type record. Those two routines make up the lower backbone of the resource

Figure 2
Uncallable Support Routines

Support Routine	Description
mlGetIndTypePtr	Returns a pointer to a type record given an index
mlGetTypePtr	Returns a pointer to a type record given a type
mlGetIndResourcePtr	Returns a pointer to a resource record given an index
mlGetResourcePtr	Returns a pointer to a resource record given an ID
mlGetResourceNamePtr	Returns a pointer to a resource name given a type and ID
mlGetResourceOffset	Finds the offset to a resource on disk
mlLoadResourceSize	Loads the size of a resource from disk

Figure 3
Sample Code

```
void PlayWithMacs (void)
{
   char Pathname[] = "MacResourceFork";
   char MacTypeString[] = "STR ";
   LongWord MacType;
   Handle Map;
   Handle MacResource;

   /* Open the Macintosh-style resource fork */
   Map = mlOpenResFile ((LongWord) &Pathname, stCString + stPointer);
   /* ALWAYS check for errors after opening the fork */
   if (!mlResError)
   {
      /* Convert a Macintosh type string to a number */
      MacType = mlStringToType ((LongWord) &MacTypeString, stCString + stPointer);
      /* Load a Macintosh resource */
      MacResource = mlGetResource (Map, MacType, -128);

      /* Use the resource any way you want... but don't forget to dispose of it! */
      DisposeHandle (MacResource);

      /* Close the open fork */
      mlCloseResFile (Map);
```

management functionality. Next at bat is the `mlGetIndResourcePtr` routine, which is sister to the `mlGetIndResource` routine. The `mlGetIndResourcePtr` routine returns a pointer to a resource record which is contained in the resource map. At the clean up position is the `mlGetResourcePtr` routine, which when given a type and ID combination, calls the `mlGetIndResourcePtr` routine repetitively until a matching ID is found and then returns a pointer to that resource record. Those two routines make up the upper backbone of the resource management functionality.

Another useful routine is the `mlGetResourceNamePtr` routine which, when given a type and ID, returns a pointer to a resource name which is contained in the resource map. The resource name functions make this call to find out what the resource name actually is.

There are only two internal routines left. The `mlGetResourceOffset` routine, when given a type and ID, returns the offset to the resource record in the resource fork. The `mlLoadResourceSize` routine, when given a type and ID, calls the `mlGetResourceOffset` routine to find the offset to the resource record, and then it loads in the start of the resource record, which is the length of the following resource. (The file mark is left alone so that a subsequent read will start reading the actual resource data. When you call `mlGetResource`, it simply calls `mlLoadResourceSize`, then it allocates enough memory to hold the resource, and finally it issues a read call to load the resource. Not much work for the routine which is the most important, you'd think. But, looking at the chain of calls that is actually made, we find that `mlGetResource` calls `mlLoadResourceSize`, which calls `mlGetResourceOffset`, which calls `mlGetResourcePtr`, which calls `mlGetTypePtr` [which calls `mlGetIndTypePtr` repetitively and then calls `mlGetIndResourcePtr` repetitively. That is a lot of work!)

Theory And Practice

Well now that you know all of the calls, you should be able to load Macintosh resources! But, since examples always carry greater weight than mere explanations, Figure 3 contains some sample source code which will open a Macintosh resource fork, perform a couple of resource-related operations, and then close the fork. Be sure that you read the comments to see exactly what's going on inside the code. The code is in C—for those of you who prefer Pascal, check out the LASERbeam source code for a real working example.

MacBlah

OK, so now that you've got all these wonderful routines to play with, what is there to do with them? Well, for one thing, you could write a PostScript® Type 1 font downloading program. Since that idea has already been taken, why not try writing a simple program that will let you take Mac resources and convert them into their logical IIGS counterparts? Or, why not write a conversion program that will translate some of the Mac's word processing files into a format that EGOed can use?

For the specifics on how to use any of the Miscellaneous Library routines in your programs, break out the **MiscLib.Docs** and related files located in the **GSP.V5.N4.SEA** self-extracting archive on your *GS+* Disk.

If you have any questions about the Miscellaneous Library, send them in! I especially want to hear any suggestions you might have for additions to the Miscellaneous Library. Putting all of these routines in one place has already made my IIGS programming easier—I hope it does the same for you. *GS+*

Glossary

In each issue of *GS+* Magazine, we present a glossary of some of the more common terms in the IIGS world and some of the more uncommon terms that we use in each issue. If you have a term or bit of jargon that you would like to see explained, let us know and we'll try to get it in a future "Glossary" installment. Also, don't forget about the glossary that's in your IIGS owner's manual! At this point, it contains many more terms than the *GS+* Glossary!

Past installments of the *GS+* Glossary can be found on your *GS+* Disk in the plain ASCII text file, **Glossary** (see "How to Use Your *GS+* Disk" for more information). Entries marked with an "*" have appeared in previous installments of the *GS+* Glossary and are repeated here for our beginning readers or because they have relevance to topics discussed in this issue.

Bit *

The most basic unit of computer data, represented by a "1" or a "0". This can be thought of as a switch which is either on (having a value of "1"), or off (having a value of "0"). See "Byte" below.

Byte

A byte is a group of eight bits that are considered as one unit. Perhaps the best way to think of a byte is to consider it as being able to hold one character. (Note that this is a vast over-simplification.) So, using this example, the string "Glossary" would take up eight bytes of memory. See also "Kilobyte," "Megabyte" and "Gigabyte," below.

D.P.I.

D.P.I. stands for "Dots Per Inch." D.P.I. is a measure of the quality of output you can expect to get from a printer. The more dots per inch a printer can produce, the finer the resolution of the printouts. For example, the ImageWriter II is a 72 d.p.i. printer, the StyleWriter is a 360 d.p.i. printer, and most LaserWriters are 300 d.p.i. Some newer laser printers are capable of 600 to 800 d.p.i. By comparison, most books and magazines are printed on devices that produce 1,200 d.p.i. or more.

Gigabyte

A gigabyte (also called a "gig"), represents 1,024 megabytes of memory. This is equivalent to 1,073,741,824 bytes.

Kilobyte

If you are familiar with the metric system, you probably know that "kilo" stands for 1,000 (i.e. 1 kilometer = 1,000 meters). However, in computing terminology, "kilo" means 1,024. Therefore, a kilobyte (abbreviated "K", as in "256K"), is actually 1,024 bytes of memory. See also "Megabyte," below and "Gigabyte," above.

Megabyte

A megabyte (abbreviated "MB", as in "40MB"), represents 1,024 kilobytes of memory. This is equivalent to 1,048,576 bytes. (See also, "Gigabyte," above.)

MIDI

MIDI stands for "Musical Instrument Digital Interface." MIDI is a series of standards covering physical connectors, electrical specifications, data formats, communication protocols, and conventions for assigning sounds to instruments and transmitting those sounds from one MIDI instrument to another.

Outline Font *

An outline font is a font whose characters are represented by a set of mathematical equations. By scaling these equations, the characters in the font can be accurately rendered at any size.

PostScript® *

PostScript is a page description language. That is, it is a language that allows you to specify the placement of text and graphics on a page (or pages). You will most commonly find PostScript implemented in laser printers, but PostScript has also been adapted for use on video displays (Display PostScript), FAX machines, and other output devices. PostScript was invented by Adobe Systems Incorporated and is included in most Apple LaserWriter printers. The complete reference to the PostScript language can be found in the book PostScript® Language Reference Manual, whichi is published by Addison-Wesley.

PostScript Type 1 Font

A PostScript Type 1 Font is a special type of PostScript language program that tells a PostScript imaging device (like a laser printer) how to draw each of the characters of a particular typeface. As each character is needed, the PostScript device runs the Type 1 Font program to create the character at the proper size. Type 1 Fonts are "Outline Fonts" (see above), which means that they can be drawn at any size without the need for a bit-mapped font for each size you wish to use. (See also "TrueType" in the Glossary file on your *GS+* Disk.)

rSoundSample

"rSoundSample" is the name given by Apple to a sound sample that is stored in the resource fork of a file. These are the types of sounds that the Sound control panel plays. However, the term rSoundSample is also commonly (but incorrectly) used to refer to a type of file (file type $D8 and auxiliary type $0003) that has the express purpose of holding one or more rSoundSamples. The proper name for this type of file is actually a "sound resource file."

RTF

RTF stands for "Rich Text Format." The RTF file format was created by Microsoft to allow word processing and page layout files to be easily moved between computer platforms while retaining as much formatting information as possible. To help ensure portability, RTF information is always stored in plain ASCII text files. To properly use the information in an RTF file, software must be written to translate the RTF information into something useful for each particular computer platform. Without this software, an RTF file will appear to be just an ASCII text file with a lot of gibberish in it.

Softdisk Issue Text

Softdisk Issue Text is a special file format that is actually a compressed Teach file (see "Teach" below). The Softdisk Issue Text format is used by the fine folks at Softdisk Publishing to get as much information as possible on each issue of Softdisk G-S.

Teach *

Teach is a simple file format created by Apple Computer, Inc. that allows font, size, and style information to be stored with ASCII text documents. The Teach format is supported by most of the newer IIGS editors, including our own EGOed NDA. For more technical information on the Teach file format, refer to the Apple II File Type Note for file type $50 and auxiliary type $5445.

GS+

GS+ Classifieds

Readers can place an ad in the *GS+* Classifieds for only $5. This cost buys 25 words in one issue of *GS+* Magazine. Additional words are just 25 cents each. The *GS+* Classifieds are a great way to contact thousands of other IIGS owners.

The deadline for inclusion of a classified ad in the next issue (Volume 5, Number 5) of *GS+* Magazine is May 6, 1994. Simply send your ad along with your name, address, phone number, number of issues to run, and payment (made payable to "EGO Systems") to *GS+* Classifieds, P. O. Box 15366, Chattanooga, TN 37415-0366; or call us at (615) 843–3988, Monday through Friday between 9 a.m. and 6 p.m. Eastern Time, to place an ad with your MasterCard or VISA. You can also FAX us your classified ad by calling our FAX number: (615) 843-3986.

Bargains! Bargains! Bargains!
Apple Memory Board (0K) - $25
256K SIMMS, 150ns (Perfect for OctoRAM & older Macs) - $5 each!
"Magnetic Media! Do not bend or X-Ray!" labels, perfect for mailing diskettes. 144 labels for $5
Free shipping on labels or if you take more than $10 worth of the other stuff off our hands!

Contact: *GS+* Magazine
P. O. Box 15366
Chattanooga, TN 37415-0366
(615) 843-3988

MusicWriter Fonts
At last, professional output from MusicWriter GS with the MW.Font package! The MW.Font package includes:
1) A TrueType version of the font, for use with Pointless.
2) A complete set of IIGS bitmap fonts, from 14 to 112 points.
3) A PostScript version of the font, for PostScript printers (you must download the font manually with a utility program).
4) A Macintosh package (PostScript, TrueType and bitmap fonts) for Macintosh network servers.

Price: $85 (by International check, includes shipping and handling).

Contact:
P. Pavlides
19 Botsari Str.
Thessaloniki Greece 546 43
FAX (31) 858 552

And More Bargains!
MDIdeas Supersonic Stereo Card with SoundSystem 2 amplified speakers - $99
Gravis Joystick - $29
$5 each:
- Software - Marble Madness, Three Stooges, Qix, Gnarly Golf, 816/Paint, DeluxePaint II.
8-bit: Labyrinth, Copy II Plus v9.1
- Books - Apple IIGS Machine Language for Beginners, Mastering the Apple IIGS Toolbox

Contact: Mathew Popoff
Box 730
Grand Forks, B.C., Canada
V0H 1H0
(604) 442-5397

Spanish Grammar
The first English to Spanish Grammar and Translator program supplements for Apple IIs. Compatible with AppleWorks' TimeOut Grammar. The "Translator (English/Spanish)" Basic - $69.95 or Advanced - $139.95 and "Spanish Grammar" Basic - $59.95 or Advanced - $129.95. Order both Basic supplements for $119.95 or both advanced supplements for $249.95, plus $5.00 per order shipping and handling. Available on 3.5-inch disks only.

Send check or money order to:
CCC Incorporated
P. O. Box 681954
San Antonio, TX 78268
(210) 344-6102

GS+ Classified Ad Order Form

Ad copy: _____

_____

_____

_____

_____

_____

_____

Number of issues to run: _____ Number of words: _____ Total enclosed: $_____

Name: _____ Phone: (_____)_____-_____

Address: _____

City: _____ State: _____ Zip: _____

GS+ Ordering Information

GS+ Magazine is published bimonthly and sold for $4.00 an issue for the magazine only, and $8.00 an issue for the magazine + disk. But, if you sign up for a 1-year subscription (six issues) or a 1/2-year subscription (three issues), you can save 11-25%! To sign up, send this completed form (or a photocopy) along with a check or money order (payable to **"EGO Systems"**), or your credit card number, to:

GS+ Subscription Services
P.O. Box 15366
Chattanooga, TN 37415-0366

Name:..... Phone: (.....).....

Address:.....

City:..... State:..... Zip:.....

Please ✔ item you wish to order:
- ❑ 1-yr subscription (mag + disk) - $36
- ❑ 1/2-yr subscription (mag + disk) - $20
- ❑ 1-yr subscription (mag only) - $18
- ❑ 1/2-yr subscription (mag only) - $10
- ❑ Sample issue (mag + disk) - $8
- ❑ Sample issue (mag only) - $4

Please ✔ delivery method:
- ❑ Third Class to U.S. - free
- ❑ First Class to U.S., or Canada/Mexico, or Foreign surface - $2.00 per issue
- ❑ Foreign Air Mail - $5 per issue

Please ✔ payment method:
- ❑ Check or Money Order enclosed
- ❑ Bill my MasterCard
- ❑ Bill my VISA

Card #:.....
Expiration Date: ...../.....
Signature:.....

If you prefer to use your credit card to order by phone, give us a call at 1-800-662-3634, Monday through Thursday, between 9 a.m. and 6 p.m. Eastern Time. (9 a.m. till 5 p.m. on Fridays.) All subscriptions will start with the next issue published. *GS+* Magazine is published bi-monthly, so please allow up to 12 weeks for delivery of the first issue of a subscription. Tennessee residents add 7.75% sales tax. *System 6, 4MB of RAM, and a hard disk are recommended for magazine + disk subscriptions.* Please remember to indicate delivery method and include additional charge if indicated.

Apple IIGS® Owners, Are You Topless?

You are? Well for goodness sake, put a shirt on! But, not just *any* shirt . . . an official *GS+* Magazine T-shirt! Yes that's right, now you can have one of these fabulous shirts for your very own. Too good to be true you say? Well it *is* true! For only $11.50 you can have your very own *GS+* Magazine T-shirt! And, they come in a variety of exciting sizes, just look:

Marvelous Medium!
Ludicrous Large!!
Exorbitant Extra-Large!!!
Really Exorbitant Extra-Extra Large!!!!

To get yours, send this completed form (or a photocopy) along with a check or money order (**payable to "EGO Systems"**), or your credit card number, to: *GS+* Magazine, P.O. Box 15366, Chattanooga, TN 37415-0366.

Name:..... Phone: (.....).....
Address:.....
City:..... State:..... Zip:.....

# of Medium Shirts:..... # of Large Shirts:..... # of Extra-Large Shirts:..... # of XX-Large Shirts:.....
❑ Bill my MasterCard ❑ Bill my VISA ❑ Check or money order (US funds only, payable to "EGO Systems")

Card No..... Expiration Date:...../.....
Signature:..... ..... Total ($11.50/shirt):.....

If you prefer, use your credit card to order by phone. Give us a call at 1-800-662-3634 (foreign orders call 615-843-3988), Monday through Friday, between 9 a.m. and 6 p.m. Eastern Time. Or, you can FAX your order to us at 615-843-3986. Tennessee residents add 7.75% sales tax. First Class shipping to the United States, Canada and Mexico, or surface mail to all other countries is included in the $11.50 price. Add $3.50 per shirt for Air Mail to all countries other than Canada and Mexico.

***GS+* Magazine**
P O Box 15366
Chattanooga TN 37415-0366

Forwarding & Return Postage Guaranteed

Bulk Rate
U.S. Postage
PAID
Chattanooga, TN
Permit NO. 616