GS+™

July
August
1991

Volume 2
Number 6

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

KansasFest!

Writer's Block

By Steven W. Disbrow

Thanks

I certainly don't want to beat a dead horse, but I'd like to take this opportunity to thank everyone that called or wrote in response to my editorial on piracy in the last issue of *GS+* Magazine. With only one exception (guess who!), everyone I spoke with was behind us all the way. It's nice to know that all of you think that what we are doing here *is* worthwhile.

Welcome Aboard

Joining the *GS+* Magazine family this issue is Wilma Tucker. Beginning with this issue, Wilma will be in charge of all of the Departments here in *GS+* Magazine. It will be Wilma's job to see that letters get answered, User Group Connection Disks get sent out in a timely fashion, and that I don't use the word "damn" very often in this column. Wilma was president of the local Mensa chapter here in Chattanooga for 8 years and she knows the English language real good.

The Merchandising Begins

If I could only pick one thing I really wanted *GS+* Magazine to do (in addition to being the best source of IIGS information), it would have to be: "Manufacture and market swell merchandise bearing the *GS+* logo." Well, now that we have cleared up that little name nastiness with Apple Computer, Inc. (see "Contest #4 Update" in this issue), my dream has finally come true! That's right, boys and girls, *GS+* T-shirts are now available! They come in a couple of different sizes and, uh, any color you want, as long as it's white. For more information on how to order these swell new shirts see the ad in this issue. (Actually, these are *very* nice shirts. We've been "test wearing" and "test washing" them for a couple of weeks now and they haven't shrunk or faded in the least.)

Joanie Loves Chachi & Apple Loves IBM

Several folks have called up to ask me what I think of the fact that Apple and IBM have signed a letter of intent to develop products jointly. First of all, it's just a letter of *intent*—nothing has happened yet. Second, if it takes as long for Apple and IBM to get the ball rolling as it did for Apple and DEC, we may all be retired before anything comes of this. However, if something does happen . . . well, the best case scenario is that Apple and IBM will develop a new computing platform that maintains total backwards compatibility (software and hardware) with the Apple II, Macintosh, PC, and PS/2. That would be very nice, but it isn't very likely. More likely is that Apple and IBM will come up with a new platform that is software compatible with the PS/2 and Mac. Except for the most basic peripherals (floppy drives, SCSI devices, etc.), backwards hardware compatibility would be lost. The *most* likely scenario is that Apple and IBM will come up with a totally new platform that is completely incompatible with everything that is available today. Whatever happens, it should give Steve Jobs and Bill Gates a few good laughs, and it should make a dandy file server.

Speaking Of File Servers

Well, if you haven't heard by now, I guess I should tell you. Noreen has sold her Atari ST and bought herself a Macintosh LC. As a favor to me, she is letting us use her Macintosh as a file server (using Personal AppleShare under System 7) so Joe and I no longer have to exchange floppy disks via "Sneaker-Net." Not to worry though, *GS+* Magazine is still *produced* on the IIGS. It's just *stored* on the Macintosh while we are working on it. If this is what Apple means by "Synergy," I like it! (Although it does cost a pretty sizable chunk of money!)

Another advantage to this setup is that we can finally test all of our programs for AppleShare compatibility. Thus far, all of our programs have checked out fine (although EGOed will be getting a tweak or two to follow Apple's guidelines in this area more closely).

While all this is very nice, I hope that there will be something like Personal AppleShare for the IIGS soon. I'd even settle for software that turns the IIGS into a dedicated file-server. After all, not everyone can afford a Macintosh LC (what the "LC" stands for, I have *no* idea), and no one that I know wants to use a Macintosh Classic as a server.

Speaking Of Macintoshes

OK. I know that, in the past, I have used this very column to denounce "computer bashing." I am especially disdainful of "Mac-bashing." It is an unproductive waste of energy that could be better directed towards making the Mac and IIGS work together better.

But, when *Bill Heineman* (author of Bard's Tale GS and Crystal Quest GS) calls you up and says he wants *you* to publish one of *his* games . . . well, that's something you just don't pass up! Actually, the game Bill wanted us to publish was his shareware game "Daleks." I felt that the BBC might not like this too much, so we kicked around a few concepts for changes to the premise of the game. After several lame ideas by Joe and myself, Noreen came up with the concept of replacing the Daleks with Macintoshes. (Of course, this was before she bought one . . . a Macintosh that is, not a Dalek.) Someone threw out the B-grade horror movie angle and "MacZombies" was born. Michael Quinn was called upon to do the title screen and Bill made the necessary changes to the program in a matter of days. The rest, as they say, is history.

Besides, MacZombies is *not* Mac-bashing—it's Mac-*smashing*!

Diz

GS+

CONTENTS

FEATURES

PROGRAMS

REVIEWS

Products marked with an asterisk (*) were review copies provided by the publisher.

DEPARTMENTS

GS+ Magazine

July-August 1991
Volume 2, Number 6

Publisher, Editor
STEVEN W. DISBROW

Associate Editor
NOREEN RIBARIC

Technical Editor
JOSEF W. WANKERL

Production Manager
SUSAN THOEMING

Departments Editor
WILMA TUCKER

Contributing Editors
DAVE ADAMS
BRIAN M. WINN
GREG ZIMMERMAN

On The Cover

A group of KansasFest attendees take a look at the latest offerings from Æ. From left to right, they are: Joe Wankerl, Brian Winn, Nory (in the background), Chris McKinsey, Nate Trost (kneeling), Derek Young, Jim Maricondo, Bill Heineman, and Jason Coleman.

GS+ is an independent publication, not affiliated in any way with Apple Computer, Inc. All references to either Apple or third party products are trademarked and should be so noted.

Opinions expressed in this publication are those of the individual authors and do not necessarily represent those of *GS+*.

GS+ is published bimonthly by:
EGO Systems
3535 Mountain Creek Road #A-17
Chattanooga, TN 37415-6734
(DO NOT SEND MAIL TO THIS ADDRESS—USE FOR UPS AND OVERNIGHT DELIVERIES ONLY!)

Subscription rates - Magazine only:
1/2 year (3 issues) - $8
1 year (6 issues) - $15
Subscription rates - Magazine + Disk:
1/2 year (3 issues) - $20
1 year (6 issues) - $36
Tennessee residents add 7.25% sales tax.
Add $1.50 per issue if you want First-Class delivery.
Canadian and Mexican orders add $1.50 per issue.
Other foreign orders add $1.50 per issue for surface delivery
or $5 per issue for Air Mail.

Send subscription orders, ads, inquiries, and address changes to:
GS+ Subscription Services
c/o EGO Systems
P.O. Box 15366
Chattanooga, TN 37415-0366
or call (615) 870-4960
Monday-Friday 9 a.m.-6 p.m. EST
pro-gsplus BBS (615) 875-4607
2400 Baud/8 data bits/no parity/1 stop bit

GS+ can also be contacted on these online services:
America Online: send mail to GSPlusDiz
or visit us online with keyword GSMAG.
Delphi: GSPlusDiz
GEnie: JWankerl
ProLine/InterNet: diz@pro-gsplus.cts.com

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

GS+ is produced on the Apple IIGS using GraphicWriter III, EGOed, AppleWorks GS, and an Apple LaserWriter IINT.

Letters

Dear *GS+*,

I enjoyed reading your article on the Allison Digitizing Software in Volume 2 Number 4 [March-April 1991]. As a user of the Allison software and Visionary GS Digitizer Board, I have to agree with you that the software and board combination produce excellent results. There are two things your readers should be aware of however. The first is that the Visionary GS requires DMA compatible memory. The second is that in my experience the color and contrast adjustments are very coarse. I solved the color and contrast adjustment problem by purchasing a Radio Shack (Archer) 4-in-1 Super Video Processor (Radio Shack Catalog #15-1276) and installing it between the video source and the Visionary GS board. The analog controls are wonderful compared to the Allison controls. The video processor allows, more or less, adjustment of detail, color, brightness, and noise. The only control I wish it had was a color correction or tint control. The price of the video processor is $129.95, which may be a little steep. But it has helped me capture human faces and cartoons with even more spectacular results than with the Allison software alone.

David Litz
Pekin, IL

Dear Steve,

I'd like to take this opportunity to compliment Jonah Stich on his fine article "HyperCard IIGS vs. HyperStudio" which appeared in the May-June 1991 [V2.N5] issue of *GS+*. I thought the article was both fair and accurate in its comparison of both products. There are, however, two points on which I'd like to comment.

The article states "HyperCard IIGS can, through the use of a stack called HyperMover, load and use Macintosh HyperCard stacks." The statement makes it sound as though the conversion of Macintosh stacks is a straightforward procedure. HyperMover does provide the basic functionality of the original stack, but a good deal of editing (several hours worth) remains to be done when HyperMover is finished.

The article later states "... for help using the program, you need to contact Apple, either through a user group connection or by a phone call." It is unfortunate that two very rich sources of HyperCard IIGS help were overlooked by the author. The Apple II Productivity Forum of America Online boasts a staff of 6 with varying degrees of experience in using both HyperCard IIGS as well as Macintosh HyperCard. My colleagues on GEnie's A2 and A2Pro Roundtables are equally experienced in using these products. Tim Swihart of Apple Computer also provides product support through these online services.

Marty Knight
Middletown, CT

Marty, Marty, Marty! Such a shameless plug for America Online! Why, if you had included the phone number that people have to call for more information (800-227-6364), I would not have even printed this letter. At least you gave equal time to GEnie (800-638-9636).

We certainly didn't mean to imply that HyperMover was magical in its capabilities. (Since that issue went to press, I've seen some stacks that were converted with HyperMover—I was disappointed to say the least.) If our copy of HyperMover ever gets here, we'll be sure to discuss it in our "HyperActivities" column.

Diz

To Diz and group:

Find enclosed my renewal for a year's worth of *GS+* (or whatever) Magazine. Keep up the good work! I think you're heading in the right direction with your fine magazine. I've currently "invested" an insane amount of money in my Apple II's (a II+ and ROM 03 IIGS) and plan in the near future on buying an HP III laser printer. Is this a bad decision? I'm relying heavily on your reviews and advice concerning accelerators and printer use.

Gary Wolfe
Wichita, KS

Well, with the introduction a few months ago of the Harmonie and Independence HP printer drivers, I would say that, yes, a LaserJet III would be a fairly good investment. For about the same money however, you can now get an Apple Personal LaserWriter NT. The last time I priced an NT, it was about $1,900 dollars and the LaserJet III was about $1,600. (These were "street" prices, not retail.) At that price difference, I would probably recommend going with the LaserWriter.

I've been planning to do a comparative review of the TransWarp GS and Zip GS, but I have an older 6.3 MHz TransWarp and it really wouldn't be fair to compare it against my 8 MHz Zip. I've been trying to get a 7 MHz upgrade for the TransWarp from Applied Engineering, but so far, no luck. So, at this point, I have no choice but to recommend the Zip. It's faster, and it's less expensive. It does, however, have a problem with AppleTalk. But, there is a public domain initialization file (called ***Zip.Talk****) that fixes this problem.*

Diz

Dear Steve,

Concerning your interview with Matt Deatherage [in *GS+* V2.N5] ... It is understandable that the truly productive Apple II's represent a small segment of the total unit population. If it were possible to look strictly at this segment, I think the 25% penetration numbers that Matt mentions are happening. It is obvious by the success of Beagle Bros., Vitesse, Seven Hills, Roger Wagner, etc., that there truly is a solid core of Apple II users.

The real solution then is not to get the existing base to purchase more software but to expand the central core of productive users, the ones who will purchase more software. To do this, several things need to happen. First, the Apple II community must stop bashing Apple Inc. and start recommending the IIGS to friends. Secondly, and more importantly, Apple Inc. needs to quit giving us reasons to bash them. They must do some major upgrading of the IIGS' features: better video, faster processor, etc. They must take the initiative to remove the stigma that Apple II's have poor quality video, sluggish performance, and no software support. Thirdly, they must market the machine. The current state of affairs with Apple's dealers is disgraceful. How some of them got their name in "The Apple II Guide" is beyond me.

It is foolish for Apple to comment on the declining Apple II market without pointing a finger at themselves as one of the leading reasons for that decline. It has been two years since Apple planned to make changes in the way they manage the Apple II. What has been done to attract new customers? What sign is clearly visible to the buying public that the Apple IIGS is a solid performer and would make an excellent choice for a home computer? If Apple refuses to market the IIGS as a viable home/business machine, there is little chance of any program finding a market regardless of how good it is.

While many diehards would like to see a 12 MHz IIGS, most of us are a little more realistic about what to expect. The next generation of IIGS should operate at about 6 to 7 MHz, support a 640 x 400 graphics mode, allow for 256 colors in the 320 x 200 mode, allow for a built-in hard drive and high density 3.5-inch drive. If just enough features were added to make the IIGS a viable alternative to IBMs in the home and if Apple would market it properly, Apple II sales would increase. This machine would be a real threat to the IBM and only a minor threat to low-end Macs.

. . . But what do I know? I'm just another one of those stupid Apple II enthusiasts! Have a nice day!

Bill Calhoun
Claysville, PA

Hi there,

. . . I am very disappointed that I am unable to find any type of "reference" material like Grolier's Encyclopedia that will run on the IIGS.

I've talked to Applied Engineering to see if I could use their IBM PC Transporter to access the CD through the SCSI port. (The Grolier's [CD] wants an MS-DOS machine [to run on]). No luck.

When I received your January-February [V2.N3] magazine, there was an article about AppleFest. I noticed a one-liner about Grolier working on an Apple-compatible version. Needless to day, it caught my eye. I tried to call Grolier in Chicago, they stated that the article was in error.

Could you please use your resources and do some type of follow-up article about either this program or any type of similar reference material that will work on the IIGS?

John Pachuta
Riverdale, IL

We here at GS+ *know of no encyclopedia program or similar reference material available for the IIGS, other than the encyclopedia services available on the various on-line services such as America Online, GEnie, Compuserve, and Delphi. A subscription to one of those would probably be your best bet—a modem is much cheaper than a CD-ROM drive. And since these services regularly update their encyclopedias, you would not have to worry about purchasing CD-ROM updates. However, if anyone out there knows of an encyclopedia program for the IIGS, write us here at* GS+ *and we'll be glad to share the information.*

Wilma

Dear *GS+*,

. . . Does System v5.0.4 correspond with GS/OS v3.3 or v3.03? I'd really appreciate a response on this.

How about some "Working with GS/OS" articles? Or, you could integrate them into the toolbox articles (i.e. show how to load a sound file with SF and GS/OS, then play it with the Sound Manager).

What ever happened to the updates for DeskColor, PreFixer and Fractals-GS? If you're ever bored you could update them . . .

Jon Larkowski
Chippewa Falls, WI

System Software v5.0.4 contains GS/OS v3.3.

A series of "Working With GS/OS" articles sounds like a good idea to me. In the meantime, take a look at the source code on the GS+ Disk. There are tons of GS/OS examples in there.

The updates for the programs you mentioned are "in the works." At this point, most of my time is spent simply putting the magazine together and keeping various tax-collecting institutions happy. I don't have much time for programming these days.

Diz

Dear *GS+*,

I too am concerned about the shrinking IIGS software and publications market. I've read it and heard it many times; for whatever reason, Apple owners in general and IIGS owners in particular just do not buy software. Publishers are cutting support, taking titles off the market, or just deciding not to port new software over to the IIGS, as the market seems to fall below the "critical mass" needed to sustain it.

It's possible that IIGS owners do buy software, but that in comparison to the huge (and ever-growing) IBM-compatible market, even if they buy in the same percentages, it's not enough. But there is surely some theft going on, and you put very succinctly when you said "software theft hurts." I admit that when I first got my IIGS a few years ago that it was exciting to "try" software from the "software fairy"—software that I didn't always get around to paying for. As I watched the MS-DOS and Mac markets prosper while the sales of IIGS software began to erode, I began to feel more and more uncomfortable with doing that and I stopped. As an aspiring writer (I'll never make a living at it, but that doesn't matter), I also began to contemplate how I would feel if someone stole my writing without paying for it. It's nice to argue that software is overpriced, should be free, etc. Unfortunately, the realities are that publishing and distribution costs are large, programmers have to pay their bills also, and that if software cannot be sold profitably, the price will either go up, or (more likely, in the IIGS market) the titles will be pulled.

I've heard the message before, and it's a good one; IIGS owners should support the programmers and publishers who still produce for the IIGS market. We should support them with our letters of encouragement, and with our purchases.

Which brings me to another point . . . when I read some of the reviews in *GS+*, I have some disturbing feelings. On the one hand, I believe in honesty on the part of the reviewers, but I wonder if we haven't reached a point when we need to temper at least the tone of some of our criticisms. In particular, I've read a number of reviews fairly critical of titles published by Orange Cherry. I must preface this comment by saying that I have not personally tried any of the titles in question, since I don't have any kids the age most of these programs seem to be targeted for. No doubt some of their titles are not perfect. But, while scores of companies are deserting the IIGS market, you have here a company which publishes titles which are GS/OS compatible, ROM 01 and 03 compatible, not copy protected, hard disk compatible, guaranteed by a lifetime disk replacement policy, and backed by free telephone assistance and well-written manuals. Further, they are bucking the trend and bringing out new titles all the time. If we're going to criticize their shortcomings, I think we can still afford to do it in the context of appreciating them for believing in the market. If companies like Orange Cherry decide that IIGS owners not only don't buy their titles but they whine about them for not having every feature imaginable (while also lamenting how long they take to load from a floppy on a stock IIGS), we'll end up with more publishers deserting the market and the growing perception that the IIGS, like the Commodore 64 I once had, is only good for playing games. (Games reviews already make up a large part of *GS+*; I have nothing against games, though I don't have much time to play them—I just don't want to see the productivity and educational software slighted. Keep the balance you have right now.)

I'm not sure how you feel about some of these comments. I'm not sure how I feel about some of them myself, but I think we have to really bend over backward to try to keep the IIGS market alive. And please tell your reviewers to stop complaining about load time from a floppy-based system. I admit it may seem intolerable sometimes, but GS/OS alone takes a fair amount of time to load. GS/OS offers a lot of possibilities; we expect programs to be GS/OS compliant. Let's not bemoan the fact that they're monsters to load. If you want quick loaders, go back to 8-bit software and put in a turbo board.

Richard S. Albright II
Marysville, PA

I appreciate your concern over the Orange Cherry reviews, but I'm afraid I can't agree. While we are extremely pleased that Orange Cherry continues to publish IIGS software, we can't simply say "these guys support the IIGS, so you should buy their programs." And we certainly can't sugarcoat reviews simply because Orange Cherry publishes a lot of IIGS software. I don't think that Orange Cherry would appreciate it and I know that our readers would not appreciate it!

For the most part, Greg Zimmerman (our Orange Cherry reviewer) likes the Orange Cherry programs, but his children just don't seem to get anything out of them. That is the acid test that Greg has to base his final recommendation on.

As for being critical of the loading times, it is true that most of our reviews do not contain this sort of information and it does seem a bit like nitpicking. However, we have to review educational programs with two audiences in mind: parents that are buying software for their children at home and educators with a class full of children. While parents buying for their own children probably won't mind a slow loading program, an educator that has a room full of children is probably going to be interested in how long it takes for a program to load. Every second that a child spends waiting is time lost. You will also notice that Greg gives load times from a hard disk. These times are much faster (of course) than the floppy-disk load times, but I think you will find very few IIGS's with hard disks in a classroom setting.

The bottom line is: if a product is not worth buying, we say so. We do this regardless of who the publisher is, how many IIGS products they make or how many ad pages they buy.

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 enclose a self-addressed, stamped envelope. Please address all letters to:

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

GS+

Programmer's Queue & A

By Josef W. Wankerl

Queue:
How can I create a file with multiple load segments? I know about Jump tables, data, and code segments, but I don't know how to format them for the linker and loader. My goal is to surpass the 64K boundary of a single load segment. I use ORCA/M.

Charles C. Bartley
Lake Havasu City, AZ

A:
Making a file with multiple load segments is not difficult. With ORCA/M, you specify a segment name on the START, DATA, PRIVATE, and PRIVDATA lines. For example:

```
MyProgram     START

              jsl InOtherSeg

              rtl

              END

InOtherSeg    START       OtherSeg
              Using       SegData

              phk
              plb

              lda Zero

              rtl

              END

SegData       DATA        OtherSeg

Zero          dc i2'$0000'

              END
```

When you link this, you will end up with two segments. The first segment is named "   " (there are ten blank spaces there—this is called the "blank" segment) and the second segment is named "OtherSeg." You can also use the KIND directive to tell the linker what type of segment to create (i.e. static, dynamic, direct page/stack.) Be sure that all references between segments use long addressing since you are not guaranteed that each segment will be within the same memory bank. In your case, since you want to break the 64K boundary, you can be assured that each segment won't be in the same bank.

Note also that you don't have to use ORCA/M to create programs that have multiple load segments. With ORCA/Pascal and ORCA/C, you can use the segment directives to name your procedures and functions. Look up "segment" in your manual's index to find detailed information on how the directive will fit into your program.

Queue:
Just how does the Finder (and the Standard File Operations tool set) know when you eject or insert a disk? Also, how does the Finder make that really cool graphic shut down alert with the black background?

Jon Larkowski
Chippewa Falls, WI

A:
I'll answer the simple question first. The Finder makes that really cool graphic shutdown alert with the black background by issuing the GS/OS `OSShutDown` call. If the graphic screen is active when the call is made, you will see that dialog. If the text screen is active when the call is made, you will see a textual version of the same dialog.

Both the Finder and Standard File use a *Notify Procedure* to know when volume mount and volume eject situations occur. The *Apple IIGS GS/OS Reference* by Addison-Wesley gives all the gory details of Notify Procedures. Also, GS/OS TechNote number 12 clarifies some information on volume mount situations when using Notify Procedures. This TechNote has been provided for you on the *GS+* Disk. (Sorry to just quote references here, but it would be a *lot* of information to present and you would be better off reading the official source, anyhow.)

Queue:
Do you know how to add a custom icon to a resource string for an Alert Window?

Jean-Patrick Hine
France

A:
Adding a custom icon to an alert string isn't very hard. However, when you use resources, things get a bit tricky. You can't directly call `AlertWindow` with the resource number of an alert string that will display a custom icon. You first have to load in your alert string and your custom icon with `LoadResource` calls, then you have to patch your alert string. Let's take a simple example of an alert string that will use a custom icon (pretend it is a zero terminated C string):

```
"51xxxxyyzz/This is an alert
string with a custom
icon./^Okay"
```

This string uses size template number 5 and a custom icon. Please read along on page 52-8 of *Apple IIGS Toolbox Reference: Volume 3* so you get the idea of how to set up the string to use a custom icon. Following the '1' (which designates a custom icon), you give a pointer to the icon data, plus a width and a height word. You can't stick a pointer to an icon in a resource because you never know where in memory it will be loaded, so you have to first load it and then set the pointer manually. Since you're setting the pointer, you might as well set the width and height words, too. This will make it easier on you to just use standard rIcon resources for your custom alert string icons. So, in order to hold place for a pointer and two words, you fill the space in the alert string template with garbage characters, in this case 'xxxx' represents space for the pointer to the icon, 'yy' represents space for the width word, and 'zz' represents space for the height word.

Now that you've set up your alert string correctly, you need to make an icon. Any icon will do. To give you an example of a routine that will display an alert string with a custom icon, I'll use C since it will take up less space. (See Figure 1, below.) You could do the exact same thing with Pascal or assembly, though. The "difficult" parts are commented to tell you exactly what's going on. You pass this routine the resource number for your alert string and the resource number of the icon you want to display in the alert window. The routine then loads in the string and the icon, patches the string with a pointer to the icon data plus the width and height words from the rIcon resource (read page E-48 of Volume 3 for information on the rIcon resource format), and then calls `AlertWindow` to display the window.

Got a question about programming the Apple IIGS? Joe knows programming!

Send those questions to:

GS+ Programmer's Q & A
P. O. Box 15366
Chattanooga, TN 37415-0366

GS+

Figure 1 - Loading And Patching A Custom Alert String Resource

```
int DisplayIconAlert (Long StringNumber, Long IconNumber)
{
        Handle AlertString;
        Handle CustomIcon;

        AlertString = LoadResource (rAlertString, StringNumber);
        CustomIcon = LoadResource (rIcon, IconNumber);

        /* The following statement will patch bytes 3 to 6 in the alert
        string to point to the icon data */

        *((Pointer *) (*AlertString) + 2)) = (*CustomIcon) + 8;

        /* The following statement will patch bytes 7 and 8 in the alert
        string to be the width of the custom icon */

        *((Word *) (*AlertString) + 6)) = *((Word *) (*CustomIcon) + 6));

        /* The following statement will patch bytes 9 and 10 in the alert
        string to be the height of the custom icon */

        *((Word *) (*AlertString) + 8)) = *((Word *) (*CustomIcon) + 4));

        return AlertWindow (0, 0L, *AlertString);
}
```

Working With The Toolbox

By Josef W. Wankerl

Part 3: The Miscellaneous Tools

Before I start explaining the Miscellaneous Tool Set, I think I need to say why I have decided to discuss this particular tool set instead of, say, QuickDraw II or the Event Manager. The reason is that "higher level" tool sets (such as QuickDraw II) rely on the Miscellaneous Tools. Because of this, it is important to understand the lower level capabilities that the Miscellaneous Tool Set provides to these higher level tools.

The Miscellaneous Tool Set cannot be described in terms of a general function that the tool set performs. It is, literally, a collection of tools that do not logically fit into any of the other tool sets. Some of the functions that these tools handle are dealing with the Battery RAM, reading the clock, interrupt control, mouse events, user ID manipulation, and vector initialization.

As stated earlier, other tool sets rely upon the Miscellaneous tools to perform low-level tasks. For example, the Memory Manager uses the user ID routines to help with the maintenance of user IDs (see part 2 of this series in *GS+* V2.N5 for a discussion of user IDs). The Event Manager uses the mouse routines to keep track of mouse events. There is rarely a time when you need to make a call to the Miscellaneous Tool Set, but, just in case, it's handy to know what calls are available.

Apple IIGS Toolbox Reference, volumes 1 and 3 do an excellent job of describing what the Miscellaneous Tool Set does, so I won't bore you with repetitious material. Instead, here are some important points that you should remember:

Working With Battery RAM

There are four Miscellaneous Tool Set routines which allow you to access the battery backed-up RAM area of your IIGS. If you only need to access one Battery RAM parameter at a time, the `ReadBParam` and `WriteBParam` calls should be used. If you need to access the entire Battery RAM area, you can use the `ReadBRam` and `WriteBRam` calls.

It's not a very good practice to modify the Battery RAM area. How would you like it if someone wrote a program that changed your border color to brown (permanentiy!) every time you ran the program? About the only time you should use these calls are when you are writing some kind of system program that allows the user to determine what the parameters will be—such as a CDev. It should also be noted that Apple Computer Inc. has already written these CDevs for you.

Yes, there are some reserved locations in the Battery RAM area—and that's what that means. *These locations are reserved! You should not use them!* A future system could use those locations for something and then your program would not work very well.

SysBeep

Perhaps the most useful routine in the Miscellaneous Tool Set (maybe even in the entire toolbox!) is the `SysBeep` call. This call simply causes the speaker inside your IIGS to beep. Why is it so useful? Mostly as a debugging aid. It can signal you when a critical section of code has been entered. It is also a call that doesn't alter anything—so if you need to make a tool call somewhere, but you don't want to do anything (for instance, if you want to break into GSBug using a tool call), this is the call for you.

GetTick

The `GetTick` call returns the number of *ticks* that have occurred since your IIGS was last turned on. A tick is defined as 1/60th of a second—which means that there are 60 ticks in a single second. Although this may seem fairly useless, it can be quite handy. If, for example, you need to write a program that will wait for a specified interval and then proceed, you can use `GetTick` to time your interval.

Whoops!

I made a few mistakes in last issue's (*GS+* V2.N5) article on the Memory Manager. First, I said, "When a block is resized, its starting location remains the same." This is only true if the block is *fixed*. If the block is not fixed, it is certainly not true that the starting location will remain the same! A non-fixed block may indeed be moved to satisfy a resize request.

Also, I seemed to imply that if a block was not locked, it could move at *any* time. This is not the case. If you have an unlocked block, it will remain in the same location unless you make a call to the system (a toolbox call or an OS call). When you make that call, memory then has a chance to move. Of course, with a high-level language, you never know when the run-time package will be making a system call, so it's best to lock your blocks before you work with them. Basically, I make it a practice to always lock my blocks before working with them. It will save lots of headaches if you do so, too. Working with unlocked blocks is inviting trouble.

Finally, it needs to be mentioned that, if you can, it's better to use the `NewHandle` tool call instead of `New()` in Pascal or `malloc()/calloc()` in C. Why? Well, these calls return *pointers* to the memory they allocate, and that means the memory is either permanently fixed or locked. The more fixed and locked blocks there are, the more trouble the Memory Manager has doing its job. So, if you can, it's probably a good idea to use `NewHandle` to get all your memory for a IIGS application. Of course, if you are writing something that has to be ported, use the language-supplied routines.

The advantage of the `GetTick` call is that it is not affected by the speed of the CPU. If you hardcode a time delay (basing the delay on the amount of time it takes for certain CPU instructions to execute) then your time delay will not be accurate when run on machines that have an accelerator installed.

So, the pseudocode for a one-second time delay might look like this:

```
Target = GetTick + 60;
while (GetTick < Target)
 do nothing;
```

Packing

There is not really very much information surrounding the Miscellaneous Tool Set routines `PackBytes` and `UnPackBytes`. If you ever need a way to compress repetitive data (such as graphic data) and it does not have to be very fast, the `PackBytes` routine will do the job. There are, however, much better ways to pack information, but that is beyond the scope of this article. The assembly source code on page 14-40 of *Apple IIGS Toolbox Reference: Volume 1* will step you in the right direction in using these routines, but there are a few points I'd like to make that will hopefully clarify some ambiguities.

To pack data, you pass the `PackBytes` routine a "pointer to a pointer" showing where the data you want packed starts. Note that this "pointer to a pointer" is *not* a valid Memory Manager handle—it is a pointer that points at another pointer, which points at the start of the uncompressed data. (See "Pointers To Pointers" below and page 39-2 of *Apple IIGS Toolbox Reference: Volume 3* for more information.) Then you pass a pointer to a length word which tells how much data to pack. Note that this is a *word* (two bytes) parameter—the maximum amount of data you can pack with one call to `PackBytes` is 65,536 bytes. Next you pass a single pointer to the buffer area you want the packed data to be placed in. The final parameter is a length word telling how big the buffer area is. If your destination buffer is correctly sized and your source data is less than 65,536 bytes long, you're done and the data is packed. However, if you have more than 65,536 bytes of data or if your destination buffer is too small, then you will need to call `PackBytes` again. Note that when the `PackBytes` call returns, the pointer to the source (unpacked) data area will have been adjusted to point at the next "packable" byte and the word containing the size of the source data will have been reduced by the number of bytes that were packed.

To unpack data you follow almost the exact procedure when you pack data. You can only unpack 65,536 bytes of data at a time. If your result buffer is too small or if the number of bytes to unpack is more than 65,536, then you will have to call `UnPackBytes` again. Note again that the call has changed the pointer to the source (packed) data area and the source data area length word.

Pointers To Pointers

Why do you have to pass a "pointer to a pointer" to `PackBytes` and `UnPackBytes`? Why can't you just pass a regular Memory Manager handle to the data? The reason you can't do this is that both `PackBytes` and `UnPackBytes` will change the data at the location that the first pointer points to. If you passed a real handle, then the call would corrupt the handle. So, instead, you should first dereference the real handle that refers to the memory you want and then pass a pointer to that. As an example, let's pretend that the variable `dataHandle` contains a valid Memory Manager handle to the real data. To correctly set a variable called `fakeHandle` to pass to `PackBytes` or `UnPackBytes`, you would code, in C:

```
dataPointer = *dataHandle;
fakeHandle = &dataPointer;
```

. . . or in Pascal:

```
dataPointer := dataHandle^;
fakeHandle := @dataPointer;
```

Queue

There are two "new" Miscellaneous Tool Set routines (first detailed in *Apple IIGS Toolbox Reference: Volume 3*) which allow you to easily maintain a Queue data structure: `AddToQueue` and `DeleteFromQueue`. The Queue routines are mainly to help out Assembly Language programmers, although you certainly don't have to use these routines to implement a queue (or any other data structure) in Assembly Language. These calls only add items to a queue and delete items from a queue. You will have to write your own routines to traverse the queue.

What Next?

The Miscellaneous Tool Set is not a complicated tool set to understand, but it forms part of the foundation for higher level toolbox calls. If you have any questions about the Miscellaneous Tool Set, or any of the other tool sets that have been covered, let me know and I'll try to answer them. Also, it looks like the language that most people want to see toolbox examples written in is C. It's not too late to change this, though. Be sure to let me know what language you would like to see future examples in. **GS+**

Watchdog

By Josef W. Wankerl

Watchdog is a Control Panel Device (CDev) that provides you with notification and interception of Operating System (OS) calls. What that means is that Watchdog patches into GS/OS and will notify you whenever a read or write call is being made! If, for example, you are the owner of an internal hard drive and are curious as to what disk activity is taking place, Watchdog will let you know. By intercepting certain GS/OS calls, Watchdog also offers you protection from accidental formatting and erasure of your disks.

Installation

To install Watchdog, simply use the Installer that is on your *GS+* Disk. For more help with installation, see "How To Use The *GS+* Disk" in this issue. Note that the next time you boot your system after you install Watchdog you will probably see some strange activity with the color of your screen border. This is Watchdog at work. For more information on what's going on, read the "Notification" section below.

Using Watchdog

After Watchdog has been installed, bring up the Control Panel NDA and select Watchdog from the list of control panel devices. (Watchdog will be near the end of the list.) In the Watchdog window, you will find a bunch of check boxes and a pop-up menu. These controls let you tell Watchdog which actions you want it to perform and which actions you want it to ignore.

Notification

When Watchdog notices that an OS call is being made, it attempts to notify you of this fact. Watchdog does this in three different ways: by changing the border color, changing the color of the apple at the top of the Apple menu, or by clicking the speaker. You select the way you wish to be notified from the pop-up menu. You can even have more than one type of notification active at once! But, at least *one* type of notification must *always* be checked.

Types Of Notification

If you have "Border Color" notification checked, your screen border will change to green when a read call is being made, red when a write call is being made, and yellow when any other OS call is made. If you already have your border color set to green, yellow, or red then you will not be able to see notification of that particular type of OS call.

If you have the "Apple Color" notification checked, the color of the little multicolored Apple in the system menu bar will change to solid green, red, or yellow as described above. This type of notification is useful only when you are running an application with a system menu bar that has an Apple menu. (Which is just about every IIGS-specific program—with the exception of most games.)

If you have "Speaker click" notification checked, the internal speaker in your IIGS will be clicked every time a call is made. Although the speaker click notification is rather annoying under most circumstances, it can come in handy in identifying what is going on. Take the Finder, for instance. If you turn on speaker click notification on all calls you will hear the Finder polling all online devices to check for a media switch. Then, if you use some other desk accessory that makes a lot of OS calls, you can tell which calls are made by the desk accessory and which calls are made by the Finder because they will actually *sound different*! Try it, it's wild!

Interception

There are four classes of OS calls that Watchdog currently intercepts: write calls, destroy (delete) calls, erase calls, and format calls. To intercept one of these calls, simply check the appropriate box in the Watchdog CDev. Whenever Watchdog recognizes that one of the calls you checked is being made, it intercepts the call and gives you a chance to tell Watchdog to allow the call to proceed or to cancel the call.

In theory, it is very nice to be able to intercept all of these calls. In reality, however, programs make a lot more of these calls than you may realize! Intercepting each and every call quickly becomes a chore. I recommend that you intercept write and destroy calls only when testing new public domain and shareware software—it might save you some headaches. My personal setup for Watchdog is notification of read and write calls via the "Apple color" option and interception of erase and format calls.

Problems?

That's all there is to using Watchdog! We've been testing Watchdog for over a month with absolutely no problems. However, we don't have all the programs and computer setups that you do. So, if you run into a problem, let me know about it by using the Problem Form supplied on your *GS+* Disk to report your troubles.

Programming Notes

Programming Watchdog was no easy task. There are many obscure tactics (and opcodes) that I used to get the job done. I will outline them here. Watchdog is actually two programs—the CDev, which was written in ORCA/C, and the GS/OS patch, which was written using ORCA/M.

C

The CDev portion of Watchdog is fairly simple. It first installs the GS/OS patch on boot (it can install it other times, too—read the "SetNewOptions" section). Whenever the CDev is open, all Watchdog has to do is check to see which control was hit and then set an appropriate flag in the GS/OS patch code. The code is fairly straightforward, and you should have no trouble figuring out what is going on from the comment blocks. The only obscure pieces of code deal with installing the GS/OS patch code and then finding where the patch code was loaded in order to set the flags for notification and interception.

InstallWatchCode

The InstallWatchCode function loads in the GS/OS patch code resource by first loading the resource and then performing an InitialLoad2 call on the loaded code to convert it from OMF (object module format) into machine readable code. I could have simply set the convert flag on the patch code resource and then installed the default code converter instead of doing the work myself, but then when the CDev was purged from memory the patch code would also be removed. By doing the InitialLoad2 call, a completely new memory ID can be assigned to the patch code and it can then stay in memory, even when the CDev gets purged! After the patch code has been installed, a named message is posted in the message center. This message contains the starting address of the GS/OS patch code. Once the message has been posted, the SetNewOptions function is called to load in the notification and interception flags and set the appropriate flags in the patch code. Finally, the GS/OS patch code is called to install the hooks into GS/OS.

SetNewOptions

The SetNewOptions function gets the state of the notification and interception flags and then sets the corresponding flags in the GS/OS patch code. First, the message posted from InstallWatchCode is checked. If the message doesn't exist, this means that Watchdog was not installed at boot time. Watchdog recognizes this situation and then calls the InstallWatchCode to install the GS/OS patch code. This means that you don't have to reboot your machine to install Watchdog. If the message does exist, the pointer to the start of the GS/OS patch code is retrieved from the message and then the flag data area is determined. The start of the patch code is shown below in Figure 1.

Figure 1

```
              bra StartInstall      ;two bytes
              dc i4'WatchdogData'   ;pointer to data area
StartInstall  ...installation code goes here...
                                    ;start of install code
```

The flags are at the start of the Watchdog data area and they are each a word. Also, remember that when finding the offset with C, the statement WatchdogData + xyz is not a simple addition, but really evaluates to WatchdogData + (xyz * sizeof (Word)).

Assembly

The real Watchdog workhorse is the GS/OS patch code. There is a lot of material to cover with the patch code. The patch code is divided into many sections. The first section is the initial installation code. The initial installation code first adds a notify procedure to the GS/OS notify queue and then calls the GS/OS vector patch installation code. Once this has been done, Watchdog has been completely installed in the system.

GS/OS Notification Procedure

The notification procedure gets control whenever a switch from ProDOS 8 to GS/OS is made. Whenever the OS switch is made, the notification procedure calls the GS/OS vector patch installation code to reinstall Watchdog. Whenever an OS switch is made, the GS/OS call vectors are reset and Watchdog needs to tap into them again. The Notification procedure is the easiest way of detecting an OS switch.

InstallMonitor

The InstallMonitor routine is the code to patch Watchdog into the GS/OS vectors. First, interrupts are disabled. This is done because if only one word of a vector patch is in place and a GS/OS call is made by an interrupt routine, things will go haywire. Next, both the GS/OS stack and inline vectors are routed to the appropriate Watchdog interception routines and the exit points for those routines are set to jump to the old vector points. The old stack vector is saved for Watchdog's use in displaying text dialogs without any notification or interception, and finally the interrupt state is restored.

Call Detection

Whenever an OS call is made (either inline or stack based), Watchdog will gain control. First the processor register is saved and then the X and Y register values plus the emulation mode is saved. The machine is then forced into full native mode. Next a check is made to determine whether Watchdog is active or not. If Watchdog is inactive then control is passed on to the next OS call handling routine.

If Watchdog is active, the call number is retrieved and the interception checking routine is called. If the call was denied then control is returned to the calling program after cleaning up. With an inline call, cleaning up consists of setting the return pointer past the call number and the parameter block pointer. With a stack call, cleaning up consists of removing the call number and the parameter block pointer from the stack. In either case, the carry flag is cleared and the accumulator is zeroed to pretend that the call was successful. Finally the emulation mode plus the X and Y registers are restored and control is returned to the calling program. If the call was accepted instead of denied then the notification checking procedure is called. Finally, the emulation mode plus the X and Y registers are restored and control is passed onto the next OS call handling routine.

Call Interception

Checking for a call interception isn't that difficult of a task. All that needs to be done is to check the OS call number against the corresponding call numbers that are to be intercepted. Actually *intercepting* the call is a bit more difficult. When a match occurs, the name of the call needs to be displayed in some sort of a dialog box. The call name can be found by simply using an index through the call checking procedure. An index to the name (which is stored in the Watchdog data area) is set before checking for the OS call number. If a match occurs then the index to the name is set. Once a call name has been found then all that remains to do is to prompt the user to allow or deny the call. If the Window Manager is active, an AlertWindow call can do all the work for us, including displaying the correct call name. To display the call name, the index is converted to ASCII form and then placed in the alert template string after "*" signifying which name to display. The difficult case to handle is when the Window Manager is inactive. In this case, I opted to display a text dialog. First the screen state is saved and a device search for the .CONSOLE device is made. The console device is opened and the top half of the dialog is drawn. Then the call name is drawn (using the index and a lookup table to find the correct name to display) and then the bottom half of the dialog is drawn. The console device is set to no-wait mode so that a return is not needed to signify the end of a read call and then one–byte reads are performed on the console device to get user input. The input is checked for a return (accept) or an escape (deny) keypress. Once an allowable key is pressed the wait mode is restored, the console device is closed, and the screen state is restored.

Note that around both the Window Manager and text forms of dialog code, the system busy flag is incremented. This insures that this section of code will not be called again while it is executing. Some interrupt tasks could repeatedly make OS calls causing the system to crash while waiting for the user to accept or deny a call.

Call Notification

Checking for a call notification isn't that difficult of a task, either. All that needs to be done is to check the OS call number against the corresponding call number that is to be notified, similar to the interception routine. Actually notifying the user of the call is a bit more difficult. Since there are three distinct types of calls (read, write, and other) it is easy to know how to notify the user. Whenever a match occurs, the color of the border is saved and the three notify routines are called (border color, apple color, and speaker click.) When an OS call is made, the busy flag gets incremented and when the call is done the busy flag gets decremented; therefore, a task is added to the Scheduler to execute when the busy flag reaches zero. When the OS call has finished executing, this Scheduler task will "turn off" the notification (restore the border and/or apple color).

Border Color

Changing the border color is a simple procedure. When the border color change procedure is called, the new color for the border is saved on the stack and the current border color register is retrieved. Only half of the border color register is used to change the border color—the other half must not be changed because it is part of a real-time clock.

To change the border color back to its normal value after an OS call is done, the default border color is retrieved using a `ReadBParam` call and the border color is set to that value.

Apple Color

Changing the color of the Apple menu is a strange task involving many steps. First the Menu Manager status must be checked. If the Menu Manager is not active then nothing further is done. Otherwise the handle to the system menu bar is retrieved. If the handle is zero (null) then there is no system menu bar and nothing else is done. If the handle is valid then it is dereferenced to get to the system menu bar flag byte. The invisible bit of the flag byte is checked to see if the menu bar is invisible. If the menu bar is indeed invisible then nothing is done. Finally, if all the above tests pass, then it is okay to change the color of the Apple.

In order to change the Apple color, the color word must first be determined. The color passed to the Apple color-changing routine is in terms of the border color and must be converted to a real palette color. Once the palette color is determined then color number one in the palettes for scan lines one to six must be changed. To do this, the entire palette is retrieved, the color entry is changed, and then the new palette is set. In 320 mode, simply setting the color at offset one in the table will change the apple color, but in 640 mode things are a bit different. With 640 mode, the colors are *dithered*. Basically this means that every four contiguous pixels that are displayed get their color from a different offset in the palette. This means that to change the Apple to a pure color, every fourth color in the table must be changed (offsets $1, $5, $9, and $D) to the new color value.

You'd think that changing the colors back to the multicolored Apple would be just as difficult. Well, surprise! It isn't. A simple `InitPalette` call does all the work for you.

Speaker Click

Clicking the speaker was the easiest of all the notification methods to implement. All that has to be done is to access the speaker location at $E0C030. Another reason why the speaker click method is so easy is because it does not need to be "turned off" after an OS call is finished.

The End?

Watchdog is by no means a simple program to comprehend, from a programmer's point of view. If you have any questions or if you would like to see something else done, let me know about it! Also, if you have any problems, fill out the problem report form on your *GS+* Disk and send it in. We've tested Watchdog extensively, but there still may be some bugs that we haven't found. I hope you find Watchdog as much fun as I have! **GS+**

MacZombies

By Bill Heineman
and the **GS+** Staff

You are the last Apple II owner on the planet. All of your former comrades have been converted by the "Committee For The Proliferation Of Macintosh Computing." (Actually, most of them bought MS-DOS machines—but they don't want you to know that.) And now, they are coming to get *you*! They want to turn you into a MacZombie!

MacZombies is a light-hearted game of paranoia and computer-carnage brought to you by the same programming genius that created Bard's Tale IIGS and Crystal Quest IIGS (among others), "Burger" Bill Heineman. In MacZombies, you are a lone Apple II owner that is being hunted down by a MacZombie death squad. Their job is to empower you . . . to death. Your job is to avoid the MacZombies for as long as possible and to smash as many of the little buggers as you can before they get you.

Installing MacZombies

The title screen of MacZombies features some very nice MIDI Synth music. To hear it, you must have the MIDI Synth tool (tool number 35) installed on your startup disk. To install the MIDI Synth tool on your startup disk, run the Installer that is on your *GS+* Disk. You can also install MacZombies to your hard disk using the Installer on your *GS+* Disk. For more information on installing MacZombies and the MIDI Synth tools, see "Using The *GS+* Disk" in this issue.

To run MacZombies, simply double-click on the MacZombies icon. The program will start and, after a few seconds, you will see the MacZombies title screen. When you get tired of looking at Michael Quinn's lovely title screen, press a key or click the mouse button.

Playing The Game

To begin smashing MacZombies, pull down the File menu and select New. A window containing the MacZombies playing field will open. On the playing field you will see two distinct objects. One of them (a small blue man surrounded by brown arrows), is you. The other objects are the dreaded MacZombies. The trick is to move your man in such a way that you avoid the MacZombies and cause them to run into each other. To move your man, click on the arrows surrounding him or press a number on the numeric keypad of your keyboard. (Pressing the 5 key causes your man to stand still and skip a turn, the other numbers move in directions corresponding to the brown arrows.)

The main thing to remember is that the MacZombies aren't very bright. They'll *always* come straight for you! If there's something in their way, they'll run blindly into it, smashing themselves to bits and leaving a pile of rubble on the playing field.

Your Tools

While the MacZombies may not be very bright, there are a *lot* of the rascals! Each time you clear the playing field of MacZombies, a new, larger group will appear to hunt you down. To aid you in your battle, you are armed with two very important weapons that you can select from the Commands menu: a Teleporter and the long-awaited Apple IIGS advertisement!

When you activate the Teleporter, you will be transported to a random location on the playing field. Since the Teleporter was a last minute hack (the inventor was turned into a MacZombie just after he built it), there a still a few problems with it. Namely, you could end up on top of a MacZombie.

The Apple IIGS advertisement is your most powerful weapon in that it will disintegrate any MacZombies that are next to you (they have to be standing on the brown arrows to be affected.) However, for all its power, you only have one available for each group of MacZombies that you face. Use it wisely.

Scoring

As you play the game, the number of MacZombies you have destroyed will be displayed at the top of the playing field. Ten or fewer MacZombies destroyed means you are a MacSympathizer. Fifty or more means you are a fairly good MacSmasher. One hundred or more and you'll get in II heaven.

The End?

Not on your life. Look for MacTerminators coming soon to your IIGS. All we have to do is make it up. **GS+**

KansasFest

By Steven W. Disbrow

Even though this is only the third year it's been held, the A2-Central Summer Conference has already become something of a Mecca for Apple II developers. In fact, it's even picked up it's own nickname, "KansasFest." Every year, Apple II developers come together for two days to attend seminars, hack code, and take abuse from Matt Deatherage. (Just kidding Matt!) It was with this same spirit that we boarded our rented van and headed for Kansas City.

We arrived at Avila College (where all of the really cool people stay during KansasFest) and began to unpack. Over the next few hours, several folks drifted by our room to chat and pick up free magazines. It was really great to finally meet so many of our subscribers face to face. For me, this was the best part of KansasFest—meeting a bunch of great people and simply shooting the breeze. The main topic of conversation was, of course, how Apple had run the Apple II into the ground and whether or not they would do anything to salvage the situation. However, the tidbits of System 6 information that Apple had provided in our conference materials made us all very hopeful that perhaps something wonderful was about to happen.

The Conference

The next morning everyone got up early and got ready for the big announcements by Apple. (If you think you know your favorite software author from talking to him over the phone, you should try sharing a bathroom with him.) As we all filed into the hall, there was a real sense of anticipation. We already knew there would be a System Software v6.0 announcement (from the tidbits Apple had already given us), would we get to see a new Apple II?

The first speaker was Uncle DOS (Tom Weishaar of Resource Central). Actually, it was a video of the speech Uncle DOS gave at the first KansasFest. It was a funny tape and did a good job of breaking the tension that all of the anticipation had built up. After Tom finished with us, he turned the mike over to the biggest name in the Apple II market, Roger Wagner.

Roger's presentation, "New Horizons For The Apple IIGS," started with a brief tour of the history of hypermedia. From its humble beginnings with the old AppleVision program, Roger led the crowd to the current state of hypermedia: HyperStudio. At this point, Roger announced the future of Apple IIGS hypermedia: HyperStudio v3.0 (see "Product Updates" in this issue). To say the least, HyperStudio v3.0 is an *amazingly* slick program that looks like it will totally wipe out every complaint I have ever had with HyperStudio. (Yes, it even has a scripting language!) As Roger demonstrated feature after feature, it became obvious that he and his company are firmly committed to the future of the Apple IIGS. After a while though, the presentation became not so much a discussion of "New Horizons For The Apple IIGS," but an ad for HyperStudio v3.0. Oh well, I do that sort of thing myself sometimes, and it *was* a great demo!

After a while, Uncle DOS pretty much had to drag Roger off the stage so that Apple could get their presentation started. After a brief wait, we filed back into the auditorium and the show started.

The first thing that was announced was the new EtherTalk card for the IIGS and IIe. (See "Rumors, Wishes & Blatant Lies" in this issue for more information.) This card will bring some serious networking speed to the Apple II and gives me some tiny hope that small business might start considering the II again—at least as workstations tied to a Macintosh server.

The next new Apple II product announcement was the long-awaited SuperDrive controller card. (See "Rumors, Wishes & Blatant Lies" in this issue for more information.) With this card, you can use Apple's SuperDrive to read and write 1.44 MB floppy disks in either ProDOS or HFS format. (HFS is the Macintosh filing system. It stands for Hierarchical Filing System.) At this point, you are probably thinking, "HFS? The IIGS can't use HFS disks!" Under System Software v6.0 it can!

System Software v6.0

The rumors about System 6 have been circulating for almost a year now and, frankly, it was worth the wait. (See "A Peek At System Software v6.0" in this issue for more complete information.) Many of the enhancements are a developer's dream come true, and the simple fact that Apple *did* it is a splendid sign that all is not yet lost.

For users, the biggest improvements should be the inclusion of the HFS FST and the ability of ProDOS 8 v2.0 to recognize more than two devices per slot (up to a total of 14 devices). The HFS FST means (among other things) that we can now buy disks of *any* size and format them as one huge HFS partition. Of course, ProDOS 8 programs won't be able to access anything stored on HFS disks, and you won't be able to *boot* from an HFS disk, so it's a good idea to keep at least one ProDOS partition around. The new ProDOS 8, with its increased device limit, should remove the problems that a lot of Vulcan hard drive owners have been having. Namely that whenever Apple upgrades the operating system, Vulcan owners have to pay Applied Engineering for a new version of ProDOS 8 that will allow the Vulcan to recognize all of its partitions.

The other major change in System 6 is the new Finder. (See "A Peek At System Software v6.0" in this issue for more complete information.) It had been rumored that Andy Nicholas (author of ShrinkIt and GS-ShrinkIt) had been doing a good job on updating the Finder, but no one really knew just *how* good a job he had done. Finder v6.0 is a great piece of work. With the exception of multitasking and a few other minor features, almost everything that you can find in the Macintosh System 7 Finder will be in the new IIGS Finder. During his presentation, Andy got several rounds of applause, he certainly deserved them!

In fact, the *entire* Apple II engineering team deserves a standing ovation. Thanks to them, our "ancient" machines can read four different disk formats, write two different disk formats, and operate under the same user interface as the most over-hyped computer in the world! It is their efforts that keep Apple II alive.

Muncha Buncha

After the Apple show was over, it was time for lunch. Now, I had not intended to write about the piddly things at this show, but I must say that the lunches served were something of a joke. The first day, we got sandwiches. The second day, we got tacos. I love sandwiches and tacos, but for a total package cost of $300, I (and everyone that I spoke with) was expecting a bit more. (The individual cost of the lunches was $7.50. I know this because Noreen had to pay that much for hers. She did not sign up for the conference, but she wanted to have lunch with us—at least the first day.)

During the Expo, we were all charged $7.50 for box lunches that contained a sandwich, pickle, chips, cookie and choice of lemonade or iced tea. $7.50? A lot of those lunches went uneaten the first day of the Expo! The second day of the Expo, the price mysteriously dropped to $6. I imagine a lot of those went uneaten as well.

Back To School

Having gotten that off of my chest (and the chests of the dozen or so people that asked me to write about it), I guess that it's time to get back to the conference itself. Since the morning sessions had gone so far over schedule, the speech that Apple II "Evangelist" Rob Barnes was scheduled to give was postponed until the next day. So, having finished with lunch, we wandered off to our conferences.

For the most part, the conferences were fairly good and there were a wide range of topics covered. Everything from preparing documentation to the inner workings of the Ensoniq sound chip was covered. The main problem is that a body can only be in one place at a time and there were always four conferences going at once. Fortunately, Resource Central had the foresight to record each conference so attendees could at least listen to what they missed.

At this point, I guess I should get something else off of my chest. While the conferences were good, I think they could have been much better. For the most part, it was simply a lack of organization and/or public speaking skills on the part of the speakers. This is understandable, these people are *developers*, not politicians! But, again, this show cost a lot of money and, more than once, I came away from a conference feeling disappointed. Perhaps the folks at Resource Central should have slightly stricter requirements for the conferences that are given. (Yes, I hear you. Next year I *will* make an attempt to get up there and try it myself!)

Our Lunch With Mr. Barnes

At lunch the next day, Rob Barnes, the new Apple II "Evangelist" got up to deliver a speech. I had heard some really good things about Mr. Barnes, and was excited to finally get the chance to hear him speak. The first thing he did was to ask us if we wanted to hear the truth or did we want the fairy tale? The truth was what we asked for and, to his credit, Mr. Barnes gave us the truth.

The truth is that Apple *still* does not seem to know what to do with the Apple II. Well, that's not exactly right. Apple knows what they want to do with the II: they want to let the II sell itself. Perhaps I misinterpreted the message, but it seems that Apple is convinced that the II will continue to sell itself without any help from them, so they are going to do as little as possible to interfere with that situation. There will be the occasional upgrade (System 6, EtherTalk, SuperDrive) and a renewed commitment to keep the II alive in the education market (the recent "The Journey Never Ends" posters and flyers that were sent to user groups and educators). But, there will be no real, sustained, attempt to recruit new developers or customers for the II.

Nory, Diz, Nuzzi, and Gerry at the Apple Central Expo

In fact, the whole speech centered around the use of the Apple II in education and how, if we wanted to survive as Apple II developers, we should either publish "subscription software" (what a great idea!) or educational software for the IIe. As support for this position, Mr. Barnes told us that 50% of the Macintosh LCs that are sold are sold along with an Apple IIe emulation board. We were also told that several years ago, sales of the Apple II into the home market mysteriously dropped off while sales into the education market remained more or less steady. Since then, education sales have maintained while the home market has all but vanished. Both with no help from Apple. No explanation was given as to why the home market sales disappeared, but it was painfully clear that Apple has no interest in getting them back.

After lunch and the speech were over, Mr. Barnes found himself in a position that (by now) is probably very familiar to him: up against the wall with a dozen or so angry Apple II people firing questions at him. To his credit, Mr. Barnes handled all of the questions that came his way, and didn't look *really* uncomfortable more than half a dozen times. To *everyone's* credit, I don't think anyone shouted at anyone else.

Several times now, I've referred to Mr. Barnes as the Apple II "Evangelist." The reason I put the quotes around evangelist is because, apparently, the definition has been changed. According to my copy of ***Webster's Encyclopedic Unabridged Dictionary of the English Language,*** an evangelist is "a preacher of the gospel" or "a revivalist." In other words, someone who spreads the word. When I asked Mr. Barnes if there were any plans to spread the word to new developers, his answer was that his job is simply to keep current developers happy. There are no plans to actively seek new Apple II developers. Mr. Barnes is an evangelist that has been directed to speak only to the choir.

Before I move on, I just want to say that Mr. Barnes handled himself very well under fire. It's clear, at least to me, that he really does care about the Apple II, its developers, its users and its future. It's just that his hands are tied.

The Bogus Journey

Having to swallow so much truth at one meal can often upset one's stomach, so a group of us decided to unwind that night by taking in the premiere of *Bill and Ted's Bogus Journey* at the local theater. Since we had a van, we got to drive a group of about 10 to the show (after asking them, "What's your favorite magazine?") Without a doubt, this was the best part of the trip. Bill and Ted provided an "Excellent!" diversion from the events of lunchtime. In fact, having seen Bill and Ted outwit Death himself, it seems that it will only be a matter of time before the Apple II community figures a way to "Melvin" Apple.

The Apple Central Expo

Having been renewed by the movie (and a few other events, one of which is pictured on our cover), we headed for the Apple Central Expo. This is the first year the Expo has been held and, for the most part, it was a very nice show. It was a small show though, only about 30 companies were in attendance. (See "Apple Central Expo Exhibitors" for a complete list.)

Like the Developers Conference, the Expo provided a great opportunity to schmooze with the people from Apple and "ordinary" folks. Best of all, it gave all the developers a chance to meet *new* people and show them their products. Although I don't have the official figures, I would estimate that during the two days of the Expo, about two or three thousand people made their way to the Expo. Many of the vendors (including myself) didn't make as much money as they would have liked, but, we all agreed that the main thing was simply getting the word out to as many people as possible.

The mood of the show was very upbeat and, for the most part, everyone was excited about being there. When the show was over, it was announced over the P. A. system that the show had been very successful and that there would probably be a repeat performance next year. With the AppleFest's disappearing, this was very good news indeed!

And Finally . . .

I just want to say thanks to Mike Nuzzi and Marty Steinberg of Triad Venture (publishers of Graphic Disk Labeler and The HyperStuff Collection) for letting us share their booth at the Expo. Thanks also to Gerry Rasmussen and Brian Winn for all of their help at the show. Finally, I want to thank Uncle DOS and all of the great folks at Resource Central for putting the Developers Conference and Expo together. All in all, KansasFest was the most fun I've had in a long, long time! **GS+**

Apple Central Expo Exhibitors

Alltech Electronics Co., Inc.
Apple Computer, Inc.
Balloons Software
Big Red Computer Club
Byte Works, Inc.
Centrex
Claris Corporation
Comp USA
Computer Covers Unlimited
Computers... ASP/Connecting Poin
Compu-Ware
DreamWorld Software
Econ Technologies
Educational Resources
inCider/A+
Interplay
Kitchen Sink Software, Inc.
LRO Computer Sales/New Concepts
Micol Systems
PC Globe, Inc.
Power Industries, LP
Procyon, Inc.
Ouality Computers
Raptor, Inc.
Resource Central
Roger Wagner Publishing, Inc.
Seven Hills Software
SSSI, Inc.
Sunset Laboratory
Three-Sixty Pacific
TMS Peripherals
Triad Venture Inc.
Zip Technology

A Peek At System Software v6.0

By the **GS+** Staff

The long awaited arrival of IIGS System Software v6.0 is near! Apple gave a demonstration of the new IIGS system at KansasFest and we got to watch. Apple hasn't *officially* announced System 6 yet, so *the following features may or may not be present in the final release*! But, you can be sure that they are real.

GS/OS Internal Enhancements

Lots of enhancements have been made to GS/OS. They are kind of hard to describe in a running narrative, so we'll just throw them all out as we come to them.

The GS/OS `.CONSOLE` device will have a major speed boost. Character I/O via the `.CONSOLE` was *supposed* to be speedy in System Software v5.0 and later, but they didn't quite get it right. This time they did.

Another problem that was corrected is the way GS/OS handles pathnames beginning with a digit. Previously, if you had a pathname such as "6 times," GS/OS thought that the "6" was a prefix number instead of part of the filename. System 6 will fix this.

A feature that has been lacking from the very first version of the System Software was the ability to detect a disk insertion during a volume mount dialog (you know, the dialog that always asks you to insert that disk you haven't seen in about a month). In the past, you would have to insert your disk and then press the return key to let the system know you had inserted it. Now you won't have to press the return key, System 6 will know automatically.

To make life easier on programmers and users alike, the `OSShutdown` call will eject all removable media when it shuts the system down, but *not* when it restarts the system.

In the old days, when a quit call was made without a user ID on the quit stack, the program that was first started would run again. In less technical terms, what this means is that if you booted your system with a floppy disk and the start program was not a program launcher (like the Finder), when you quit the start program, it would be run again! This gave the illusion of an "endless loop" and was quite annoying. With System 6, the old "Program Launcher" program (a simple Standard File dialog that allows you pick a new program to run) has been built into the system and it will take control if you quit and there is not another program launcher around to take control.

A new call has been added to GS/OS in order to make it easier for users to name files using the different naming conventions of the various FSTs that GS/OS now supports. The `JudgeName` call will verify the syntax of a file, folder, or volume name. Optionally, the `JudgeName` call can convert the name to conform to the proper syntax for the file system.

ProDOS Great!

In System 6, the functionality of the **BASIC.Launcher** program has been built into GS/OS and P8. What does this mean? Not much really, except that you won't need to have the **BASIC.Launcher** file on hand if you want to launch a program that uses ProDOS's **BASIC.System**. Finally, System 6 will come with ProDOS 8 version 2.0. The main difference between ProDOS 8 v2.0 and the current ProDOS 8 version is that version 2.0 will handle more than two devices per slot, up to a maximum of 14 devices total. This one got a big round of applause when it was announced!

Enhanced

The next few GS/OS enhancements will mainly be of interest to programmers. GS/OS has been enhanced so that the `ChangePath` call can now rename volumes that have duplicate names. The `GetName` call now returns a user ID as well as the name of the current application. This will allow your application to always be able to find its path, even if prefix 1 and 9 have been changed. The `GetRefInfo` call now returns the resource number and the file level that a file was opened at. The `Volume` call has been extended to return device characteristics (including media write protected and read-only FST bits), the device ID, and the volume name—basically everything that a `DInfo` call returns and that the current `Volume` call doesn't. Finally, the `Format` and `Erase` calls now have a minimum `pCount` of 1. If you don't supply more than one parameter, the system will try to figure out the rest of them for itself. The `Format` call will allow the user to edit the volume name and will use the new `JudgeName` call to inform the user of the correct syntax for a volume name.

FSTs

The feature that sets GS/OS apart from every other operating system in existence is its ability to use multiple file systems seamlessly. With System 6, users will get three new File System Translators (FSTs)! The first, and perhaps the one most requested, is the HFS FST. HFS is the file system used by Macintosh computers. Under HFS, you can have volumes of just about any size and you don't have to partition them into 32 MB chunks! That's right, you can buy that 210 MB hard disk you've had your eye on and use it as one big disk. Just because you can now use Macintosh disks in your IIGS, doesn't mean that you can run Macintosh programs though. The HFS FST only allows you to use the more flexible HFS filing system (how would you like to have file names that were 32 characters in length and had spaces in them? You can do that under HFS!) to read and write to Macintosh disks.

The remaining two FSTs are the DOS 3.3 (Apple DOS 3.3, not MS-DOS 3.3) and Apple Pascal FSTs. These FSTs are read-only.

Only the technical junkie would care, but, in System 6, the ProDOS FST has been modified to support HFS style option lists. That's so HFS style directory information can be stored in the key block of an extended file. This means that if a file is copied from a HFS disk to a ProDOS disk and then back to a HFS disk, all the information for the file will remain intact.

Drivers

System 6 will also come with a few new device drivers. If you have a RAM disk, there is a new device driver that speeds up access to your RAM disk. If you thought it was fast before, just wait!

If you have a SCSI card, you will be able to install drivers to let you use SCSI scanners and SCSI tape drives.

If you want, you can install the new System 6 StyleWriter printer driver, and use that groovy new StyleWriter printer that everyone has been clamoring for. Note that TrueType is *not* supported on the IIGS. That's a Macintosh-only goodie.

While the DOS 3.3 and Apple Pascal FSTs were in development, Apple found out what the rest of us have known all along: accessing 5.25-inch drives is amazingly slow and nasty. So, the Apple 5.25-inch disk driver was rewritten so that it will be restartable from memory and 200-300% faster! Wow!

Archiver

System 6 will ship with a utility to perform GS/OS file and volume backups and retrievals. The Archiver will be able to copy entire volumes (only the blocks in use or all the blocks) or a collection of files and folders from a single volume. Files and folders can be selected individually or by a selection criteria based on filenames, dates, backup bits, and a whole host of other parameters. Backups can be made to floppies, hard drives, tape drives, and even RAM disks. Also, the Archiver will compress the data, but only during full volume backups.

Teach

System 6 will also ship with an application called Teach. Teach uses the TextEdit tool set to allow the viewing and editing of text files, Installer script files, and Teach files. Teach files (previously called TeachText files) are special text files that retain text style information (such as bold and italic.) Additionally, Teach can import AppleWorks 3.0, AppleWorks GS, and MacWrite 5.0 word processor files! Although Teach isn't a revolution in word processing, it means that, as a developer, you will be able to assume that everyone that has System 6 also has something that will read text files and Teach files. So, you can ship your documentation as a Teach file with no worry of someone not being able to read it.

Universal Access

For people with handicaps, using a computer may not always be possible and at best, it's cumbersome. Apple realized this and has included a few utilities in System 6 to make it easier for handicapped people to use the IIGS.

The first two are called mouse keys and sticky keys. With mouse keys you can use the keypad to move the mouse and click the mouse button. With sticky keys, you can press modifier keys (like option and shift) one at a time instead of all at once.

System 6 will also come with CloseView. CloseView is a program that will magnify parts of the Super Hi-Res screen to 2, 4, 8, and 12 times their normal size. For people who can't see very well, this is indispensable. It should also prove very popular with graphic artists that need to do lots of detail work.

Finally there is the Video Keyboard. The Video Keyboard is a window that is always in front of all your other windows. It has a representation of the IIGS keyboard on it, and when you click on one of the keys, it pretends that you actually typed that key on the keyboard. If you can't type for some reason, but you can move a mouse and click, this lets you use the keyboard.

Installer

The Installer program has also been rewritten. For installing System Software there will be a one-button install. Simply click one button and away it goes to install your new System Software. If you are not installing System Software, you can use a custom installation. Custom installations are similar to the way the Installer currently works. New Installer scripts will have their own file type (so you can double-click on them and run the Installer), will be volume independent (meaning that you don't have to hardcode the name of a volume in your script), and can be run from any folder, not just the **Scripts** folder. The Installer scripting language will be changed, so if you want to develop scripts for use with the new Installer, you might want to pick up the new version of IIGS TechNote #64 when it becomes available.

Finder

System 6 will ship with a greatly improved version of the Finder. It is modeled after the Macintosh System 7 Finder and a large percentage of IIGS System 6 Finder has been implemented using resources. This means that if you want to change how parts of the new Finder look, you can pull up a resource editor (like Genesys) and use it to change the new Finder to your heart's content.

The window scrolling speed has been improved and is now roughly eight times faster than the current Finder. The speed-up is most notable in windows where you are viewing By Name, By Date, etc. The time to open folders and to clean up a window has also been improved. When you select a group of items and drag them to a different location, only the outlines of the visible selected items will move. Currently, outlines of all selected items are animated and it takes a long time for all of that drawing to get done. And yes, the feature you were waiting for is almost here, you will finally be able to drag folders out onto the desktop!

The new Finder will support a feature called "tunneling." You invoke tunneling when you open or close a folder by holding down the option key. As the folder opens, the window of its parent folder closes behind it. This way you can "tunnel" down to a folder nested deep within your hard drive without leaving the screen cluttered with the windows of all the folders parents. Reverse tunneling allows you to close the current window and be placed back into the window of the parent folder. Even if the parent folder was already closed! Finally, if you hold the option key down while clicking on the close box of a window, the new Finder will close all of its windows on the screen.

Another feature in the new Finder's windows are pop-up menus showing the path that you have to take to get to the folder that a window represents. Simply hold down a few modifier keys and click on the window's title and you will be shown a pop-up menu detailing the path to get to that folder. If you want, you can select a folder from the menu and it will be opened for you!

The Impossible Drag

Has this happened to you? You have a small window overlapping part of a bigger window and you want to drag some items from the big window into the small window. Under the current Finder, the big window moves to the front (covering the smaller window) the instant you click on it. This is what Apple calls "the Impossible Drag." Finder v6.0 solves this problem by not moving the newly selected window to the front until after you release the mouse button.

More Great Stuff

A new option in the Item Info window will let you actually edit comments attached to files. You will even be able to add comments to your old files! However, adding a comment to a file also adds a resource fork to it, making it unusable with ProDOS 8 programs. The new Finder will warn you of this before it lets you add a comment.

The new Finder has "hooks" in it so that DAs, Inits, and Control Panels (formerly called CDevs) can now talk to it. Using these hooks, DAs, Inits, and Control Panels will be able to do a multitude of things inside the Finder that simply were not possible before. For example, we should soon see a New Desk Accessory that will open itself and display Super Hi-Res pictures when you click on them from inside the Finder.

The new Finder will use a different type of Icon file. Icons will be kept in resources. Not to worry though, the new Finder can also use all of your old style icons and it should convert them internally to the new icon format.

Applications can now contain a new resource (called the `rBundle`) which tells the new Finder what types of documents that particular application can open (among a host of other things).

Of course, there are many more Finder additions and enhancements, but these are the major ones.

Toolbox Changes

In order to keep this article from filling the entire magazine, we'll just hit the major enhancements to the toolbox in System 6. If you aren't a programmer, these would be almost impossible to elaborate on, if you are a programmer, no elaboration should be needed. So, let's just take them in rapid-fire fashion.

First, a new tool set has been added: the Media Control tool set. This tool set provides a way for your applications to seamlessly interface with multimedia devices such as CD players and VCRs.

The Control Manager now handles thermometer controls so that you can easily make and use thermometers just like the Finder. If a control can be a target it will have a frame around it if it is the current target. List controls, for example, can now be target controls and can receive keystrokes to select items from within the list.

Edit line controls will scroll right and left if the contents are larger than the visible region. This will be very helpful for typing in those 32 character file names under the HFS FST.

A `FindRadioButton` call has been added to the Control Manager so your application doesn't have to do the work of finding out which radio button is currently selected.

In System 6, Command-W *always* closes the front window.

NDAs and CDAs are sorted alphabetically in their menus, but this option can be turned off if you wish. You can now navigate through the CDA menu using keystrokes! NDAs can have more than one window open at the same time.

The Pixel Map and Fake Modal Dialog routines have been incorporated into the toolbox so now your applications can easily have I-beam cursor support (among other things).

Menus will be able to have icons in them.

A new Miscellaneous Tool Set call, `WaitUntil`, lets your application wait for a certain amount of time to pass no matter what speed your computer is running at.

The Resource Manager will have better support for named resources.

And finally, there will be key equivalents for buttons inside Alert Windows.

Worth Waiting For?

All these "facts" about System 6 may seem just dandy, but what does it mean to you? Hopefully it means that when System 6 is released later this fall, you will be able to get more done with your IIGS in a shorter amount of time. Hopefully, it will also mean that developers will take another look at the IIGS and begin to exploit some of the powerful new features that the Apple II engineers have worked so hard to give us! **GS+**

GS+ Back Issue Information

Sep-Oct 1989 (V1.N1)
- $4.50 mag • $5.50 disk • $6.50 both

- System Software 5.0 Compatibility Chart
- NoDOS - A file utility New Desk Accessory complete with ORCA/C source code on disk
- Graphics Galore - Drawing "how-to" with 3 pictures on disk
- Reviews of Arkanoid II (new custom levels on disk), Crystal Quest, ORCA/C, Rocket Ranger, Silpheed, Test Drive II, TransWarp GS, Turbo Mouse ADB
- PLUS: Graphics, rumors, and the most over-hyped product of the year!

Nov-Dec 1989 (V1.N2)
- $6.50 disk (magazine is sold out!)

- EGOed - An NDA text editor (TML Pascal II source code on disk)
- Brush with Greatness - Tips on drawing faces (pictures on disk)
- PLUS: Original icons and new levels for Laser Force on disk

Jan-Feb 1990 (V1.N3)
- $6.50 disk (magazine is sold out!)

- Rotator - A beginner's desktop programming tutorial and program w/source code written in ORCA/C
- Winning Arkanoid II Levels
- Brush with Greatness - Space graphics (pictures on disk)
- HyperStudio stack version of GS+ V1.N2 on disk.

Mar-Apr 1990 (V1.N4)
- $6.50 disk (magazine is sold out!)

- All About Control Panel Devices - with Desk Color CDev and ORCA/C source code on disk
- Random IIGS Programming Notes - An EGOed update
- Brush with Greatness - Architecture on your IIGS with pictures of the CitiCorp building and Frank Lloyd Wright's house on disk

May-Jun 1990 (V1.N5)
- $4.50 mag • $6.50 disk • $9.50 both

- AppleFest Report
- Beginner's Guide to System Disks - Part 1
- GS/OS prefixes - PreFixer CDev and ORCA/Pascal source code on disk
- Brush with Greatness - How your IIGS makes colors
- Reviews of CMS SDRM 45 Megabyte 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, a sneak peek at the ZipGS

Jul-Aug 1990 (V1.N6)
- $4.50 mag • $6.50 disk • $9.50 both

(Only a few left—call for availability!)

- KansasFest Report
- Beginner's Guide to System Disks - Part 2
- Transfusion - An NDA terminal program (ORCA/C)
- Reviews of AMR AS800K 3.5-inch drive, Salvation: The Exorciser, Disk Access, MD-BASIC, Katie's Farm, Task Force, BLOCKOUT, OMEGA, 2088: The Cryllan Mission, Hunt for Red October, Revolution '76, Where in the U.S.A. is Carmen Sandiego?

Sep-Oct 1990 (V2.N1)
- $4.50 mag • $6.50 disk • $9.50 both

- Brush With Greatness - making the most of your digitizer
- Interview with Brian Greenstone (programmer of Xenocide)
- PING - video table tennis program (Merlin assembly)
- Shuffle - an Init file that allows you to move desktop windows from the foreground to the background (ORCA/M)
- Battery Brain - CDev saves BRAM parameters to disk (ORCA/C)
- Reviews of GS Sauce memory card, Salvation: Wings, World GeoGraph, Orange Cherry Talking Schoolhouse series, QIX, Solitaire Royale, InnerExpress

Nov-Dec 1990 (V2.N2)
- $4.50 mag • $6.50 disk • $9.50 both

- Interview with Bill Heineman (programmer of Dragon Wars)
- Beginner's Guide to System Disks - Part 3
- LaserWriting - a guide to using an Apple LaserWriter with the IIGS
- Christmas Buyer's Guide
- TeachText Translator - import and export TeachText files in GWIII
- Reviews of Quickie Hand Scanner, AE 3.5" Disk Drive, Salvation: Renaissance, USA GeoGraph, Rastan, Captain Blood, HOSTAGE, Questmaster, Pipe Dream, The Immortal, PIRATES!

Jan-Feb 1991 (V2.N3)
- $4.50 mag • $6.50 disk • $9.50 both

- AppleFest/Long Beach '90 and the Apple II Achievement Awards
- Interview with Jim Carson of Vitesse, Inc.
- Introduction to System Software v5.0.4
- RAM Namer - a CDEV that can rename your RAM disk at boot time, with ORCA/C source code on disk
- *GS+* program updates - Battery Brain v1.1, EGOed v1.32c (now written in ORCA/C), GWIII TeachText Translator v1.1
- Reviews of ZipGSX, LightningScan, Design Your Own Home, Print Shop Companion IIGS, Your IIGS Guide, Dragon Wars, 2088: The Cryllan Mission - Second Scenario, Space Ace, Sinbad and the Throne of the Falcon

Mar-Apr 1991 (V2.N4)
- $4.50 mag • $6.50 disk • $9.50 both

- Interview with Dave Hecker of Seven Hills Software
- Working with the Toolbox - Part 1: The Tool Locator
- Quick NDA - an init that can assign control-keypad equivalents to your New Desk Accessories, with ORCA/M source code on disk
- The New Order - a NDA that can reorder the contents of your directories, with ORCA/C source code on disk
- *GS+* program updates - EGOed v1.33, Transfusion v1.1.1
- Reviews of Harmonie, Independence, InWords, Allison Digitizing Software, MAX/Edit, Software of the Month Club, Super GS Award Maker, Talking Speller II, Halls of Montezuma

May-Jun 1991 (V2.N5)
- $4.50 mag • $6.50 disk • $9.50 both

- Interview with Matt Deatherage of Apple DTS
- Working with the Toolbox - Part 2: The Memory Manager
- Autopilot - A program launcher with ORCA/M source code on disk
- Softlock - A password protection NDA with ORCA/C source code on disk
- *GS+* program updates - EGOed v1.34, NoDOS v1.5
- Reviews of TMS Pro R45 Removable Hard Drive, RamFAST/SCSI Card, HyperCard IIGS vs. HyperStudio, McGee at the Fun Fair, Talking Classroom, Talking Multiplication and Division, Bouncing Bluster II, Space Shark, Transylvania III

All prices include $1.50 postage and handling (orders will be sent First-Class to the U.S., Air Mail to Canada and Mexico, and surface to all other countries. For Air Mail to all other countries, add $5 per issue). Tennessee residents add 7.25% sales tax. Mail back issue requests to: *GS+* Back Issues, c/o EGO Systems, PO BOX 15366, Chattanooga, TN, 37415-0366; or call (615) 870-4960, Monday through Friday between 9 a.m. and 6 p.m. EST, to 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 card number, expiration date, and signature.

A Conversation with Roger Wagner Part 1

Believe it or not, Roger Wagner of Roger Wagner Publishing, Inc. (publisher of HyperStudio) got his start in the Apple II world in 1978 (it's hard to believe that Apple has been around that long)! Below are some of the things he talked with us about in our "dorm room" at KansasFest. He had so many interesting things to talk about that we can't fit them all in this issue. We will run the rest of the conversation next issue.

GS+ A lot of attention is focused on HyperStudio these days. What different types of products are you developing, or are all your new products going to be designed to complement HyperStudio?

RW HyperStudio is sort of a focal point or center of gravity for us, but as for being a HyperStudio company . . . it kind of depends on how you measure it. If you mean, are we going to be a one-product company and only have HyperStudio and that's it—no. If you look at our catalogs, we have about 40 products, and that makes HyperStudio one-fortieth of our line. The general company charter is to be *the* educational multimedia company, so if somebody comes in and says "I've got a really neat routine for controlling CD-ROMs," or something like that, we're interested in it. People create stacks that have an educational value. It doesn't have to be in a school setting, it could be for people at home. Having fun with a computer is an educational process; it could be a stack about programming in assembly language.

The second part is, not everybody feels like they can write educational software. But, any time you convey an idea from your mind to someone else, you can say that education has taken place. So things that just share ideas are, in a way, still educational. But for the people that are really the hackers—and I consider myself kind of in that category, when I sit down to write programs, I like to write utilities and things that invert bits, rearrange data, and do something neat—for somebody creating some other project that I maybe can't imagine, I think there's a lot of opportunity for developers everywhere. I think of HyperStudio v3.0 as "slotted" software, because it's not just that you can create clip art: between the extras, the transitions, the "new button actions," and all the different places you can start putting things in—you can pretty much pick the level you like to play on the IIGS and say "well, where does that fit in?" It's not like AppleWorks or Carmen Sandiego. Imagine trying to build a developer base for Carmen Sandiego or AppleWorks—they're nice programs, but they're somewhat narrow. HyperStudio is more of an environment in and of itself, and it's not a bad prototyping environment. Why write 5000 lines of code to turn on the menu bar, initialize QuickDraw, and start up the ACE tools just to test your little beep routine? With HyperStudio v3.0, you can write your beep routine, stick it in HyperStudio, try it out and then be right back in your assembler re-assembling it and be through with it. This is wandering a little bit from the original question, but yes, HyperStudio is going to be the center point. And there are so many ways that people are going to find something they enjoy—I think there are a lot of opportunities.

GS+ How would you interpret what Rob Barnes, the "Apple II Evangelist," said at the A2-Central Developers' Conference?

RW I thought he was pretty straightforward in what he could say and what he did say. There were a number of different messages—he talked about "the days of writing a bad word processor and getting rich are over" and that's true. And a lot of programmers are still thinking "I've written this really cool something, and all I have to do is get it listed in magazines, and I'll bet lotsa guys will want to buy it." We have this little joke: George and Mark Lotsa in Iowa—they're the two guys that buy these things, the "Lotsa guys." But after you sell just two copies, you're in trouble. I think the market for software isn't just installed customer base, because you can say there's thirty million Commodore 64's out there, but if they're all in closets, and people aren't looking for the latest releases every week, it really doesn't matter.

And that's the same problem with the IIe—there's a lot of IIe's out there, but if you go in the schools, and find the people who are the sparkplugs, the real energizers and motivators in their school building, more of them have the IIGS's, and they're looking for the things that the technology can really do. Otherwise, Apple IIe hypermedia products should be outselling HyperStudio 4 to 1. So I've approached it very pragmatically "if that's good advice, then maybe I should write a IIe version," but I don't think I would sell four times as many of those as I do HyperStudio.

GS+ The message I got from Mr. Barnes was that, basically, he was telling everybody to write educational software for the IIe—and that was it.

RW Well, I've addressed the IIe part. The educational part, I think, is unavoidably true, because the market has self-selected itself. If you took a look at the registered owners of Apple II's, you would find that a large percentage of them were educators. Apple said that three years ago, home market sales evaporated, but educational sales stayed pretty much the same—there was very little loss in educational sales. Apple [II] sales went from, oh, I'm just speculating, $1 billion to $500 million—the home side dropped, the education side maintained. Part of the keynote presentation I gave mentioned that.

If you came and visited Earth and asked people "what is the computing experience?" and did that every two or three years, starting in about 1978, the answer as to what computers were all about would have changed radically from visit to visit. In 1978, everyone with a computer did programming. Programming was fun and a challenge and it made the computer do things. Then a little later VisiCalc came along, and computers were great computational tools that let you model a "what if" situation, and that's why everyone felt they should own one. What happened to programming? Nobody was doing programming anymore. A couple of years later, it was telecommunications. Computers would open up the world through telecommunications networks, the computing experience became defined by modems.

Imagine you were a software company, and you're going from games to telecommunications to word processors to desktop publishing. If you lock yourself in and say, "I'm digging my heels in—I think computers are telecommunications," then you're going to discover eventually that modem programs are hard to sell. So if you today wrote a modem program, you wouldn't find many people to sell it to, and you'd be really depressed. But you can't blame the world for not maintaining its interest in what you're interested in. Education is what the world is interested in doing with Apple II's, and I think what's changed is that if we had had this conversation a year ago, it would be about how to write a better "Oregon Trail," or how to teach kids how to tell red triangles from blue triangles. With HyperStudio, that spectrum has opened up by orders of magnitude.

Two years ago, if somebody said, "list educational software publishers," would Roger Wagner Publishing have been listed? I don't think so. Did we have some massive change, did we start putting up school posters and chalkboards in our office—whatever it is that educational software publishers do? No, we're the same company, we think exactly the same way, it's just our products are being drawn into the educational market, because that's where we make the sales.

GS+ Although it seems impossible, some people don't know how you got started with the Apple II. Tell us how Roger Wagner Publishing got its start.

RW The quick history is that in 1978, roughly, I was teaching high school—junior high school math, science, physics, chemistry, and stuff, and that's when computers started appearing on the covers of *Popular Science* and *Popular Electronics*. I think Pet, actually—the Commodore Pet—was the one that was getting all the exposure and so I just thought that it was something I wanted to have. Actually, I shouldn't say that. I thought for a year, I was trying to decide whether to get a stereo, a motorcycle, or a computer. I was saving up $700 (that was how much the Pet cost) and with that I could also buy a used motorcycle. Believe it or not, one of the reasons I bought the computer is that I thought that if I got the other things, I would have to keep spending money on them, and if I got a computer, I would just plug it in, and I'd just type, and all I'd have to buy was electricity, so it would be the least expensive.

So with that brilliant foresight into the world of computers, and business, I decided to just buy a computer for fun and I went into a computer store. And just to show you that these problems aren't new, I just went into a Computerland and said, "I've got the $700 in my pocket, and I want to buy a Pet, so show me the computer." And he said, "Well, actually, that one is out being shown at a school somewhere today, so we can't show that to you, but we do have an Apple." And I said, "Well, OK, what is an Apple, and why should I buy it?" And he goes, "Well, the keyboard's bigger." And I said, "OK, how much does it cost?" And he said, "Well, it's like $1400." And I said, "Well, gee, you know, $700 is kind of a lot to pay for a big keyboard."

But what also struck me at the time was I felt like I worked really hard to save up this money, and I might even have been able to have been talked into going back and saving up another $700, but the salesman basically told me, "There's the computer," and went away, just like he couldn't care less.

My first reaction in that computer store in 1978 was, "I'm going to spend a thousand dollars and the salesman doesn't care?" So I was kind of offended by that. But in the little piece of paper they gave me it said "Apple Computer dealerships are invited." And I thought, "Ah ha, I'll bet I know how this works—if I could buy like two or three of these computers, all I'd have to do was convince two or three of my friends to buy one, and we'll call up saying we're a dealer and buy three, or whatever their minimum order is, and I'll get it for like the $700 the Pet cost." Everybody said the Apple was really the best computer—that's what the cool people got. You could afford to get a Pet, but the real computer to get was the Apple, since that was the one that was really good. I thought, "This is great, I'll be able to get the good one and only spend as much as the other one."

I called up Apple Computer and said, "I want to be a dealer." And they said, "Oh, that's wonderful. We'll be down—just tell us where your store is." And I said, "I don't own a store. Can't I just buy them? Maybe I'll take them door-to-door." And they said, "No, no, no, we have very strict standards." And I believed them. Actually, I did get some advice, because this time whoever I talked to was actually a local distributor—at one point, Apple went through local distributors—and he said, "Well, tell you what, what you should do is go and find an electronics store, or a computer store that sells some other brand, get them to be an Apple dealer, and just tell them you'll pay them 5% over invoice for their trouble. That way they get to be an Apple dealer, they'll get their name listed places, and you never know, someone might actually walk in and buy one. They don't have to put up any money, and they pick up 5% for their trouble of handing you the box." This sounded pretty good, so I actually went around to different places. There was a place called Computer Metrics, and the salesman there basically thought the Apple was kind of a toy, with the color and stuff, and that the Exidy Sorcerer was really the computer that he felt was going to take off. He was busy doing his stuff, and he didn't want the Apple dealership. Eventually, through some different changes in management, at one point they were the biggest Apple dealer in the country, because they were doing mail order. (That was one of the big mail-order controversies taking place.)

We were originally Southwestern Data Systems, and I picked that name because I wanted something that sounded bigger than "Roger's Computer Stuff," and I wanted something that wasn't quite as grand as "Cosmic Computer Company" and I wanted something generic so that if I changed my mind next week, I could sell printer paper or software—"Southwestern Data Systems," who knows what they sell? That was the name we had for four or five years. Originally, I eventually did find this store, I got the computer, and this was during the summer, and I started getting books on BASIC and writing BASIC programs.

Really, the only problem with this whole thing was I thought we'd kind of start this business, but I knew nothing about business or computers. I'd never taken a computer class, and certainly not business. So I got a suit, and a briefcase, and I got my little Apple II carrying bag. Then I literally went down to shopping malls and went to the tailor, and the laundromat, and said, "This is a really cool machine—can't you just imagine the potential?" And I showed them AppleVision. And for some inconceivable reason, they just didn't fork over $1,000 and buy AppleVision for their laundromat. As a result, I had a whole lot of time to sit at home and write BASIC programs. At some point a friend of mind said, "Gee, now you've written this little BASIC renumbering program—I bet the people in the user group might actually pay money for this." I said, "Nah," and he said, "You could probably get ten bucks!" And I said, "Ten bucks for a piece of software?" So my first program was a renumbering program for AppleSoft.

At that time, you could run an ad for $15 or $20 in Micro magazine. And darn if people didn't start sending me mailboxes full of checks. Of course there were some basic ratios of business that I don't know about, like you've got to charge at least six times your manufacturing cost to stay in business, so I was ending up—by the time I was done—making these things for $7 and charging $10, but by giving the dealers a 40% discount, I wasn't making any money. Anyhow, we started writing programs, and at some point I thought, "If I keep writing these, I'm going to have to keep being clever. And there's a lot more guys out there that are a lot more clever than I am, I can't be the smartest programmer out there forever." When it was started, Southwestern Data Systems (the original name of Roger Wagner Publishing) was kind of a big deal software company, because it was the only software company. But as more companies started coming on, I thought, "Gee, I can't keep this up forever." And I had this friend, Bill Blue, who had written a program called ASCII Express. And I said, "Well, tell you what, why don't I publish that for you?" So we agreed upon what was probably an unwise royalty rate for me, and we published that for a number of years.

That was what really made Southwestern Data Systems a company, as far as the name. And I think at some point, around '84 or '85, we decided it would probably might be a good idea to incorporate, but Southwestern Data Systems was already taken. There's actually a phone number you can call in the state of California, and give them three names at a time to do a name search, so I gave them Micro Works, Data Utilities Inc.—I gave them every name I could think of with Byte, Computer, Works, Digital—and they had all been taken. I decided nobody would be silly enough to call their company Roger Wagner Publishing, and darn if I wasn't right on that one! That name was available.

Over the last twelve years, the company itself has continually changed as we followed that process I mentioned earlier: the evolution of utilities, games and productivity software. In fact, at one point, we actually had a game called Bizarre that was the only game that was actually from the point of view of aliens—where the Earth people were the bad people, and the manual and and the instructions on the screen were all in hieroglyphic script.

In fact, I believe that we have a legitimate claim—I think the date was 1981—that we were the first company that ever did a game where apple- or control-w would put up a spreadsheet display while you were playing the game. That idea occurred on a hang-gliding trip. We had this party where all the software publishers went down to Mexico, on the sand dunes, and learned hang-gliding for two days. Wozniak, the Carlstons from Broderbund, the Magic Window people—kind of like a KansasFest 1981—all the people went down there. I remember talking to Gary Carlston and he was talking about how his accountant liked playing these games, but the problem was people came into the office and he really felt guilty. And I said, "I know, we'll put up a spreadsheet!" So that was like '81 or something, and there are no claims to the contrary. Everything else kind of pales in comparison. [laughs]

GS+ What's the deal with the ties? How did it all get started? (Just a little fun question. . .)

RW That's why we're here, just to have fun, and that's the deal with the ties—to have fun. Some people wear a hat, or I could get a Mohawk haircut, or dye my hair different colors, but I'm not quite that adventurous a guy. Actually, there's a thing I look for in ties, I say "Is that a HyperStudio tie? Is that bright colors?" It's just a feeling, like if you went into a restaurant with a certain decor, and they had something completely out of character sitting there, like an antique furnace sitting in a 1950's rock-n-roll restaurant. I thought it would be fun, if I have to wear regular clothes. (I can't have neon lights on my pants and shirt and be taken seriously, there's so little left.)

Actually, I think what happened was I bought one tie because I liked it, and wore it to a show. Somebody made a big deal about it, and the next day they were upset because I didn't have a tie that matched, so I had to buy a second one for the next show. Then I went to a show and they started complaining because I wore the same one twice, so now I feel like Princess Di—I have to have a new one every time I go to a show. **GS+**

Next time Roger talks about the other programmers associated with HyperStudio and offers advice to Apple II developers.

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 $10 (plus $1.50 shipping and handling) you can have your very own *GS+* Magazine T-shirt! And, they come in a variety of exciting sizes, just look:

Marvelous Mediums!

Ludicrous Larges!!

Exorbitant Extra-Larges!!!

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: ____________________

Number of Medium shirts ________ Number of Large shirts ________ Number of Extra-Large shirts ________

Total amount enclosed ($10 per shirt plus $1.50 per shirt shipping and handling) ________

❑ Bill my MasterCard ❑ Bill my VISA ❑ Check or money order (US funds)
Card No.: ________________________________ Expiration Date: ____/____
Signature: ________________________________

If you prefer, use your credit card to order by phone. Give us a call at (615) 870-4960, Monday through Friday, between 9 a.m. and 6 p.m. EST. Tennessee residents add 7.25% sales tax. Add $1.50 per shirt for postage and handling. Add $1.50 per shirt for delivery to Canada or Mexico. Add $1.50 per shirt for surface mail to all other foreign countries. Add $5 per shirt for Air Mail to all other countries.

The Molehill

By Josef W. Wankerl

In this Molehill, I'll discuss some ways in which people debug their programs. This is in continuation to the last Molehill (in *GS+* V2.N4) in which I talked about the Nifty List and GSBug debugging aids.

The first debugging tip comes from Michael Lutynski via America Online:

If you're familiar with Softdisk G-S's *OmniMouse CDA, you know that [with OmniMouse installed] if you press the shift and option keys at any time, the underlying text screen is shown. This makes it convenient to look at the* `printf()`*'s because they don't have to be output to a GrafPort. However, this only works if you don't use ORCA's* `startdesk()` *function because* `startdesk()` *patches in all* `printf()`*'s to the current GrafPort.*

For quick programs, I'd recommend using the `startdesk()` function, but for any serious work, it's better if you start the tools yourself by using the new `StartUpTools` tool call. In addition to the OmniMouse CDA, you can also use GSBug to view the text screen. Simply enter GSBug and then type "off" at the command line to turn off GSBug's debugging screen and display the normal text screen. Of course, you can type "on" to turn the GSBug screen back on. If you're really stuck for a utility to view the text screen while debugging, let me know and I might whip up a quick one for the next Molehill.

The second tip comes from Jonah Stich via America Online:

A cool way to debug things that you can't use GSBug on (PIFs and TIFs that get loaded before it, for example) is to skip around that code in the file (i.e. put a RTL at the beginning of the code) so that it's not called when the system is started. Then break into GSBug (a `_TaskMaster` *tbrk is really nice) and step through the code at your leisure. Also I've found that GSBug can be used to debug non-assembly files. You need only a minimal amount of assembly knowledge—just look for tool calls, know that a=error when c=1, and keep looking at the screen (*press *"s"* when stepping or tracing*) to see what's going on. I find you can get a pretty good idea of what the program is doing.*

Well, you can use GSBug to debug PIFs, TIFs, and CDev boot procedures! Simply make sure that the **GSBug.Init** file is before any of those programs in your **`*:System:System.Setup`** folder. You can use The New Order (*GS+* V2.N4) to accomplish this task. As for debugging non-assembly files, yes, I do that lots. It really helps if you have a copy of your source code printed out (or if you have a spare computer, load it into an editor so you can make corrections to your code when you see problems) so you can refer to it to see what's going on. Come to think of it, it helps to have a copy of your source around no matter what language you program in!

Now, back to your original idea of skipping the code of your PIF/TIF—yes, that would work. You neglected to mention how to find the entry to your code once you've entered GSBug, though. I would use Nifty List to search for memory ids of type Axxx (Nifty List command A000i). But remember, once you've started debugging your TIF/PIF, that's about all you can do with your system. If you don't remember the program counter where you broke, you won't be able to go back. Also, the program counter, stack, data bank register, and direct page for your host application will be used by the initialization file you are debugging, so they may not be in a state where you can resume your host application. Putting GSBug first in your startup sequence will alleviate this problem.

Tips

Although the following three tips aren't necessarily related to debugging, they seemed useful enough to print. All of them were sent to me by Jonah Stich.

If you have to push two coordinates for a tool call, it's always H then V, so you can just `PushLong ThePoint` *instead of pushing the two words one at a time.*

That's true. However, the `PushLong` macro evaluates to two separate `pha` (or `pea`) instructions. You really won't gain anything in performance time when your program runs. You might, however, lose some program clarity in your source code.

If you're using GSBug, the MEM registers are great places to store values, even if you don't really care what's at that memory location. When stepping through code, you can put the start address of the code in a MEM register so that you won't forget it and won't have to write it down. Same goes for handles you don't want to forget, etc.

I used to use this trick all the time until command history was finally added to Nifty List. With Nifty List, I can stick my current handles on the command line and hit control-s to save them in the history without actually making Nifty List execute a command. Later, when I want to play with my saved values, I can pull them up quickly. Additionally I can add some kind of comment to the line (not so with MEM values) so I know exactly what I'm looking at. Plus, because I'm in Nifty List, I can then do anything I want to with that value—disassemble what's there, dereference it if it's a handle, and much more!

If you use a directory sorter like ProSel or Wings to sort your ***2/AInclude*** *folder, you can put the most used M16 files at the beginning and things like ADB and ACE at the end. Speeds things up a bit when macgening.*

I'd never thought of that! But it makes lots of sense. I think I'll use The New Order from *GS+* V2.N4 to sort mine!

If you use a debugging technique that hasn't been discussed here, I'd like to know about it, and it might even go into another future Molehill. If we use any of your items, we'll either send you a snazzy new ***GS+*** T-shirt or extend your subscription for one issue! Be sure to tell us whether you want the T-shirt or the extra issue. If you want the T-shirt, don't forget to include your T-shirt size. Happy debugging! GS+

Rumors, Wishes & Blatant Lies

By Prof. G. S. Gumby

No New II

Well, we've heard it all before, and it looks like things haven't changed. According to a speech Rob Barnes (the Apple II "Evangelist") gave at KansasFest, there are no plans to produce a new Apple II CPU at this time. Oh well, it was a nice dream.

EtherNet

However, something that wasn't a dream is that Apple announced that it is working on *EtherNet* adapters for the IIGS and IIe! If you don't know what EtherNet is, well, let's just say that it's what the "big boys" use to build some serious computer networks.

The big advantage of EtherNet is its speed. Whereas AppleTalk could be compared to Congress trying to agree on a budget, EtherNet could be compared to Congress approving a pay raise for itself. It's *that* fast!

Our sources tell us that the reason Apple is giving this capability to the II is that when most schools build an AppleTalk network, they use a Mac Classic as the file server (they don't want to buy Macs to begin with, so they buy the cheapest one they can). To get technical for a second, this setup is what we call *very, very slow*.

So, it appears that Apple will now try to sell schools the Macintosh LC (which is about twice as fast as the Classic) as the file-server along with EtherNet cards for the LC and all of the II's in the classroom.

Beyond its use in the classroom, EtherNet for the II also means that Apple *might* just be ready to try selling the II to small business again. I doubt it though.

Look! Up Around $300! It's...

SuperDrive! That's right boys and girls, Apple has finally announced a SuperDrive adapter card for the IIGS and IIe. Basically, this card will plug into your computer and act as a controller for Apple's High Density (1.44 MB) SuperDrive. This will be just the thing to accompany the new HFS FST in System Software v6.0. No prices were announced, but I should think that $500 for both the card and a single SuperDrive would be a good estimate.

Sluggo

Well, unless you are locked in a very tiny, poorly-ventilated room, you have probably heard about the new Super Nintendo. Bill Heineman heard of it about a year ago and he's been working on a development system for it. It's called "Sluggo" and it attaches to your IIGS. You use your IIGS to write and compile your Nintendo software and then Sluggo downloads it to the video game system for you to test.

But, Sluggo can do more than just Nintendo software! According to Mr. Bill, Sluggo has a whole family of separate "personality modules" that will let you write software for just about any video game system you want! The only thing Sluggo doesn't do is get you the necessary licensing forms!

Sluggo should be available any day now . . . we're just waiting for Bill to send us an official press release and price list.

The Merchandising Continues...

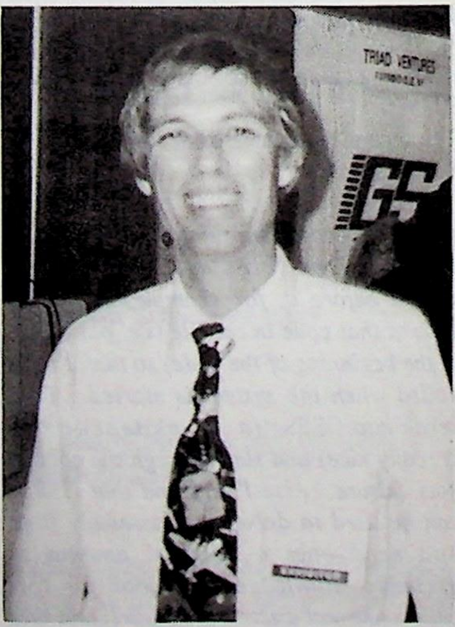
Due to the runaway success of the new *GS+* T-shirts (get yours today!), the publisher has decided to expand the items available in the *GS+* Magazine merchandise catalog. In the next few months, you'll be seeing ads for *GS+* Mugs, *GS+* Pencils, *GS+* Matchbooks, and a complete line of fully posable, *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.)

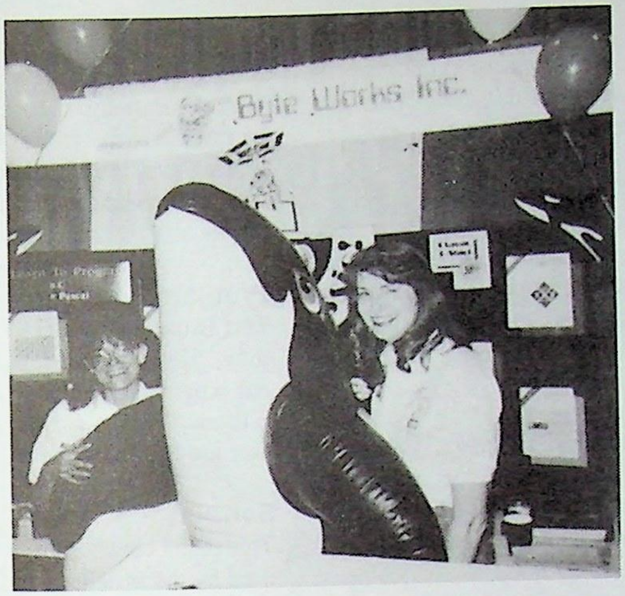
The Ties That Blind

When you say "Roger Wagner," what do you think of? Hypermedia? HyperStudio? Hyperglycemia? Well, you *should* be thinking neckties!

In a marketplace where every other word is "gloom" or "doom," (and occasionally, "Armageddon") the unusual neckties of Roger Wagner shine as a beacon of hope, happiness, and a quick way to fill a few inches in any convention report. Towards that end, we here at *GS+* Magazine proudly present our first Roger Wagner Necktie Report: A day by day accounting of the ties that Roger wore at KansasFest and the Apple Central Expo. (Before we begin, it should be noted that Roger wears only *genuine* neckties, *not* sissy clip-ons.)

Wed. July 17: Ice cream cones. Yummy!
Thurs. July 18: No tie today. Roger was wearing his "HyperStudio World Tour" T–shirt. But, we suspect he had a tie on underneath it. It was probably a really loud one too.
Fri. July 19: Everyone that looks at this tie sees something different, so, we called it Rorschach.
Sat. July 20: Airplanes and Baggage tickets. As shown on this page.
Sun. July 21: Screaming Yellow Zonkers.





Shocking Development!

In a surprising turn of events, Mike Westerfield (developer of ORCA/Everything) underwent a radical *species change* operation just before KansasFest! Mike, shown above with KansasFest Babes Barbara Allred and Patty Westerfield, at their booth at KansasFest, told us that he underwent the operation, "because I thought it would be a good promotional stunt. Maybe I should have just bought an inflatable whale instead."

A 10,000 gallon tank had to be quickly constructed so that Mike could lead the various seminars he was scheduled to conduct. Even though very little was said about programming (for some reason, the focus of the seminars shifted to things like "how stupid Dolphins are," "jumping through hoops—why you shouldn't do it" and, "balancing beach balls on ones nose"—at one point, Mike even launched into an ugly tirade against Japanese whaling practices), Mike's sessions were generally regarded as the most entertaining, and certainly the wettest. At one point, Uncle DOS and Jay Jennings of Resource Central jumped into the tank, crawled up on Mike's back and dazzled the crowd with "Two Dozen Dumb ORCA Tricks."

Lookee Here!

What's this? An ad for the Apple II? *From Apple?* My gosh. It *is*! It's like one of those slick, fold-out flyers that you pick up at your local Macint . . . er, I mean, Apple dealer. Goodness. And look inside here . . . Apple is selling a HyperCard IIGS bundle that includes a hard disk (with System Software v5.0.4 and HyperCard IIGS already installed), an Apple High-Speed SCSI card and a fully loaded Apple 1 MB memory board. Gee, if you plugged all that stuff into a ROM 03 machine, it would look just like the rumored IIGS Plus (except for the enhanced video). Gosh, I seem to remember back when Apple replaced the ROM 01 with the ROM 03, they started giving away Apple 1 MB memory boards with ROM 01 machines so they would be comparably equipped. It's a pity they had to send this to the printer before they decided to can the new II. I mean, stuff like this is what gets nasty rumors started!

Æ Singles

As you can see from our cover photo, and as we reported in this very column a few issues ago, Applied Engineering has decided to start providing a slightly different type of service . . . just kidding. Actually, the newspaper everyone is gawking over is an entertainment magazine, local to the Kansas City area, whose main focus is personal ads. Someone at the show (Tony Diaz, I think) found them at a local supermarket, noted the similarity to the Applied Engineering logo (this particular Æ stands for "Arts and Entertainment") and brought back an armful to pass off to unsuspecting Festers. As they say in *TV Guide*, "Hilarity Ensued."

MD Headed?

We hope someone out there is working on an Apple II interface to Sony's new 2.5-inch read/write compact disk technology. These tiny disks hold as much music as a regular 5-inch CD and, best of all, you can *record* on them! The technology is called MD (which stands for "Micro-Disk" I think), and hopefully, it will bring the cost of CD storage down to a point that normal humans can afford.

Wake Up!

We need your help! Yes, *your* help! Send your Apple IIGS rumors, wishes and blatant lies to us via:

America Online, Delphi:
GSPlusDiz

GEnie:
JWankerl

ProLine/InterNet:
rumors@pro-gsplus.cts.com

US Mail:
GS+ Rumors
P. O. Box 15366
Chattanooga, TN 37415-0366

If we use any of your items, we'll either send you a snazzy new *GS+* T-shirt or extend your subscription for one issue! Be sure to tell us whether you want the T-shirt or the extra issue. If you want the T-shirt, don't forget to include your T-shirt size. **GS+**

How To Use The *GS+* Disk

The first thing you need to do is **make a backup copy of your *GS+* Disk with the Finder!!!** Next, put the original in a safe place. If you are having a problem making a backup copy, give us a call at (615) 870-4960. 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 v5.0.2 or later (preferably v5.0.4), and then 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 *GS+* Disk. (From the Finder, you would double-click on the Installer icon.). When the Installer window appears, select the update you want to install from the left-hand window, and the disk you want to install it on in the right-hand window. Then click on the Install button. For more information on how to use the Installer, refer to your IIGS owner's manual.

Small Talk

Beginning this issue, we will be using GS-ShrinkIt to compress the source code on the *GS+* Disk. (If we had not done this, MacZombies and its source code would have taken up almost two disks!) To extract the source code from their archives, you will need to use GS-ShrinkIt v1.0.4 or later. If you do not have GS-ShrinkIt, check with your local user group or give us a call here at *GS+* Magazine and we will help you locate a copy.

For More Help...

Remember, each folder on the *GS+* Disk contains its own **a.Read.Me** file that describes the contents of the folder in detail.

What's On The Disk

There are 10 items in the root directory of this issue's disk. They are:

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 out about it in this file. This is a plain text file.

HyperActivities

This folder contains the Address Book stack discussed in this issue's "HyperActivities" column. It requires HyperCard IIGS v1.0 or later to operate. It can be installed in any folder on any disk.

EGOed

This folder contains EGOed v1.34. EGOed is a New Desk Accessory text editor that allows you to read and print ASCII, AppleWorks (Classic and GS) and TeachText files.

This is the same version of EGOed that appeared in the last issue of *GS+* Magazine. It is being provided in this issue so that our new readers can easily read the files on this disk. This folder also contains complete documentation for EGOed v1.34 in the file **EGOed.Docs**.

EGOed must be installed on a startup disk.

Icons

This folder contains the Finder icons discussed in the "Icons" article.

Installer

This is the Apple IIGS Installer. Run it to install the other programs on this issue's disk. For more information on using the Installer, refer to your IIGS owner's manual.

MacZombies

This folder contains the game MacZombies as discussed in the "MacZombies" article. Also this folder contains the MIDI Synth tools (**TOOL035**) and the Merlin source code for MacZombies (**Zombies.SRC.SHK**).

To use the source code, you must first extract it from the **Zombies.SRC.SHK** file with GS-ShrinkIt v1.0.4 or later and you must assemble it with Merlin v4.09 or later.

MacZombies and the MIDI Synth tools have separate Installer scripts and should be installed separately. The MIDI Synth tool set must be installed on your startup disk while MacZombies may be installed on any disk and inside any folder.

Problem.Form

This is the standard *GS+* problem report form. If you have a problem with one of our programs, fill out this form and send it to us. This is a TeachText file. You may use EGOed to view it.

Scripts

This folder contains all the scripts that are used by the Installer in order to automate the installation process.

Watchdog

This folder contains the Watchdog Control Panel Device that is discussed in the "Watchdog" article. Also in this folder is a GS-ShrinkIt v1.0.4 archive called **WatchdogSRC.SHK.** It contains the ORCA/C and ORCA/M source code for Watchdog.

To use the source code, you must first extract it from the **WatchdogSRC.SHK** file with GS-ShrinkIt v1.0.4 or later.

Watchdog must be installed on your startup disk.

Writers.Guide

This is a TeachText file that explains what you need to do in order to write reviews, articles, programs, etc. for *GS+*. You may use EGOed to view it.

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! **GS+**

Icons

By Steven W. Disbrow

This time, we have just a few new Finder icons for you, all of which were done by our very own Noreen Ribaric.

First of all, there's a dandy new icon for MacZombies. If you use the Installer on your *GS+* Disk to install MacZombies, this icon will automatically be added to the disk you that you install the game on.

Second, there's an icon for the game Reach For The Stars (reviewed in this issue). To use this icon, copy it out of the **Icons** folder on your *GS+* Disk and into the **Icons** folder of the disk you run Reach For The Stars from.

The next icon is for the Address Book stack discussed in "HyperActivities." If you use the Installer on your *GS+* Disk to install this stack, this icon will automatically be added to the disk that you install the stack on.

That's all the icons for this time. If you have any icons you would like to share with us, please send them in! **GS+**

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

DISKLESS?

If you did not receive the disk with this magazine and have decided you would like to have it, just send a check or money order for $6.50 to:

GS+ V2N6 Disk Offer
c/o EGO Systems
P.O. Box 15366
Chattanooga, TN 37415-0366

Or call us at (615) 870-4960, Monday through Friday between 9 a.m. and 6 p.m. EST, to bill it to your MasterCard or VISA.

Tennessee residents add 7.25% sales tax.
Prices include $1.50 for First-Class delivery to the U.S., air mail to Canada and Mexico, or surface to all other countries. Add an extra $3.50 ($10 total) for air mail to all other foreign countries.

Advertisers Index

By Steven W. Disbrow

Byte Works Inc. - Page 33

What can I say about the Byte Works that won't get me sued? No, seriously, these are some of the nicest folks in the business. If it wasn't for them mentioning us in the "Byte Works News" a couple of years ago, *GS+* Magazine might have disappeared after only one or two issues.

This new program, Talking Tools, includes everything you need to put synthesized speech in your IIGS programs. It even comes with a text editing program (complete with source code, of course) that can read your text files back to you!

So far, I've only noticed two problems with the Talking Tools package:

1) If you want to use them in your programs, you have to license the Talking Tools from First Byte (the original authors—anyone remember SmoothTalker GS?), *not* the Byte Works.

2) When I fed Dave Adams' defense of HyperStudio (from *GS+* V2.N5) to the talking editor and had it read it back, it crashed about half way through it. Too bad, Dave . . .

GS+ Magazine - Page 24

This is a definite conflict of interest, but what the heck! These shirts are great! Order yours today! Seriously though, if you want a size that we don't have shown, let us know. If we get enough requests, we'll have some shirts printed up in those additional sizes.

Learning Experiences - Page 48

It seemed kind of silly to continue to use a RGB monitor on our BBS, so I began to hunt for a monochrome monitor to use instead. Our good friend Greg Zimmerman put me in touch with Rick Slone and the rest of the fine folks at Learning Experiences. A few days later, a Macintosh IIsi box arrived at my door. Ripping open the box, I found that not only does Learning Experiences have great prices, they have a great sense of humor too! The monitor has worked flawlessly and I couldn't be happier with it.

ProDev Inc. - Page 42

The ProDev DDT16 is a *hardware* debugger. The hardware is well designed, but the software leaves a bit to be desired. For complete information, see Joe Wankerl's review in this issue of *GS+* Magazine.

Raptor, Inc. - Page 51

The products advertised here are *image enhancement programs* for your black-and-white 320-mode graphics. Although I have not had much time to play with the review copies that Raptor sent us, they seem to do exactly what they promise—and they do it very well! And, since our last issue was published John Majka of Raptor contacted us to say that they have sped up the operation of both programs! If you need the ability to perform the same kind of processing that NASA does on satellite photos give the folks at Raptor a buzz, there simply isn't any other product like this for the IIGS.

TMS Peripherals - Back Cover

If you need a hard drive, I *highly* recommend that you buy it from TMS. Several months ago, I desperately needed a 45 MB removable to do backups with (see review in *GS+* V2.N5). TMS got it to me in less than 24 hours and I paid about $100 less (shipping included) than the "base sticker price" of just about any of the big name 45 MB removable drives advertised in the MacRags. Since then, the drive has not given me a single problem. Even if the drive *had* been bad, TMS has a toll-free support line, a 30–day Money Back Guarantee and a 2–year Warranty!

Part of our job here at *GS+* is to find the best people in the business and point them out to our readers. I was so impressed with TMS's products and service that I practically *begged* them to advertise with us. These guys are some of the best and they truly deserve your business. **GS+**

Errata

In last issue's comparison of HyperCard IIGS and HyperStudio, the memory requirement for HyperStudio is listed as 768K. This is incorrect. The minimum memory requirement for HyperStudio is 1 MB.

In Jonah Stich's review of the HyperStuff Collection, we completely forgot to give contact information for the publisher! The HyperStuff Collection is published by:

Triad Venture, Inc.
P. O. Box 12201
Hauppauge, NY 11788
(516) 732-3771
(516) 360-0797

Our sincere apologies to the Marty and Mike at Triad for this oversight. **GS+**

ProDev DDT16

Designed by Chuck Kelly

Retail price: $189

ProDev, Inc.
P.O. Box 162
LaSalle, MI 48145-0126
(313) 241-2786

Reviewed by Josef W. Wankerl

At First Glance

The ProDev DDT16 is a hardware debugging card for the Apple IIGS. (DDT stands for Development & Debugging Tool, but it bears a nice resemblance to the outlawed pesticide.) ProDev also makes a virtually identical card for the IIe, but I will be reviewing the IIGS card from a IIGS developer's point of view. The card will fit in any slot in your IIGS and you must set that slot to "Your Card" (with the control panel) in order to do any debugging with the card. The manual that comes with the DDT16 is informative and provides examples of everything the card can do.

Starting It Up

To begin your debugging with the DDT16, you must first initialize the card. You do this by going into the Monitor (from BASIC, type `CALL -151`) and then type `Cn00G`, where n is the slot that your DDT16 card is in. Once the card has been initialized you can use it to debug. I found that initializing the card, while not difficult, definitely took out some time where I could be developing. But, for the lazy, there is an initialization file available on America Online and other information services that will automatically initialize the card when you start GS/OS. The version of the DDT16 that I have didn't come with any such initialization file, but I talked to Chuck Kelly (the developer of the DDT16) at KansasFest and he told me that the DDT16 is now shipping with a similar initialization file.

Breaking And Entering

A break point is a signal to the card that the program currently executing should be interrupted and the debugging software should be run. The DDT16 supports several types of break points. The first kind is an "implied" break. When you are using the DDT16 to step and trace through some code, the DDT16 looks at the program counter for each instruction. If the program counter matches a location where an implied break is set, the DDT16 stops. You can also tell the DDT16 how many times the address must be reached before it will break. For example, you can tell the DDT16 that you want the instruction at location `05/FD42` to be executed five times before it breaks. This is handy for checking the ends of loops or seeing what happens to your subroutine the second time it is executed.

The second kind of break is a "real" break. When you tell the DDT16 that you want to set a real break, it modifies the instruction at the given location to be a `BRK` instruction. The advantage of real breaks over implied breaks is that you can be executing the code in real time instead of being in step and trace mode. As with the implied break, you can tell the DDT16 how many times the address must be reached before it will break.

The third, and most interesting, kind of break is a "hard" break. The hard break is unique to the DDT16. It causes a break whenever a particular memory location is accessed. For example, you can set a hard break on location `02/3D22` and the DDT16 will get control whenever the instruction at that location is run or if someone reads or writes data at that location.

The fourth type of break is a "conditional" break. A conditional break is like an implied break except that the break only occurs when a condition is met. For example, you can set a conditional break that will occur when the value in the accumulator is greater than $5.

The DDT16 has a nice, long cable coming from it with a button on the end. Whenever you press this button, it tells the DDT16 to generate a Non-Maskable Interrupt (`NMI`). This causes a fifth kind of break. If you ever need to interrupt the program and start debugging, a simple press of the button drops you into the DDT16's debugging software.

Step And Trace

The most important use of a debugger is its ability to show you what is going on inside the computer an instruction at a time. The DDT16 allows you to write a small subroutine which is called after each instruction has been executed. The subroutine can test all kinds of things (the example in the manual was testing to see if location `$3002` had reached value `$80`) and then tell the DDT16 if something interesting had happened.

Another useful feature of the DDT16 is its ability to count the number of cycles used to execute code. This is really handy if you have to optimize a routine for speed. Counting cycles by hand is tedious and very error prone.

When stepping through some code, you often run across something like `LDA ($01,S),Y`. Where exactly is the data coming from? The DDT16 helps you with this calculation. It displays the effective address and its contents for each instruction when you are stepping through some code. This is, perhaps, the most useful function that the DDT16 performs.

Of course, the DDT16 has commands to execute subroutines in real time and to execute instructions in real time until a `RTS` instruction is encountered, and skip the current instruction while you are stepping through some code.

I/O Redirection

Perhaps the keenest feature of the DDT16 is its ability to redirect its output through the internal modem port of the IIGS. Using this feature you can see the results of your program on one screen and the DDT16 output on the screen of a remote terminal. A remote terminal can be another IIGS or a simple dumb TTY terminal. You can also give the DDT16 commands from the remote terminal. Also, if you want, you can direct the output of the DDT16 to the IIGS printer port so you can get a printout of what's going on.

Helpful?

The previous information may sound wonderful for debugging, but just how useful is it? And, how easy it is to use?

The answer is . . . not very. Although the card does all this wonderful stuff, it doesn't do it in a way that's friendly to the programmer. For example, when you step through some code, you go one instruction at a time. To get to the next instruction, you have to type "TR" and press the return key. Then, the current instruction is executed and the next instruction is displayed on the screen. You can't look ahead to see two or three instructions in the future. Also, for each instruction executed, you get a listing of all the registers and their values. This makes it very difficult to determine exactly what is going on. The screen gets cluttered really fast.

The Shoot Out

Although I hate to do this, I think it's only fair to point out some of the flaws in the DDT16's manual (and flyer) where it describes what makes the DDT16 better than the others. I'll put the DDT16's claim in italic and my replys to that claim in plain type. Please note that the DDT16 claims are perfectly valid when working with a *IIe*, but I'm only touching on the IIGS version and what these claims mean for a IIGS developer.

Your program may simply be too large to allow room for the debugging program. (Solution? None!)

On the IIGS, this point is not as relevant as it is on a IIe since the typical IIe has 64 to 128K. Memory is cheap these days (cheaper than the DDT16, that's for sure). To do any serious development on the IIGS, you need *at least* 1 MB of memory.

Your program may use the same area of memory as the debug program does for program or data storage. (Solution? Move the debug program, if possible, or, modify your program to suit the debugger. What a pain!)

If your program uses the same memory as the debugging software on a IIGS then your program is simply written wrong! The IIGS features advanced relocation of your program and data.

Debug programs are erased by system reboots. (Solution? Never, press the "Control, open-apple, Reset" keys simultaneously.)

...and with the DDT16, don't forget that you have to re-initialize your card (unless you have the initialization file.) It's about as difficult as reloading another debug program. Also, with debugging programs such as GSBug, it is loaded in automatically at boot time.

If your program crashes it may wipe out the debug program. (Solution? Write bug free programs that work correctly the first time.)

There is some merit to this point. But . . . what are the odds that you really do trash the debugging program? There's a lot of memory out there to mess up. And, if you are trashing the program, there *are* other means of debugging programs than by using a hardware/software debugger. Once you get that bug out you can resume debugging with your software debugger. By thinking that you can debug your programs by using one debugging method is rather narrow-minded. (This is not to imply that the people who make the DDT16 are narrow-minded, though—in fact, quite the opposite!)

There is no way to get control of a program that is running in real-time. (Solution? Use the ProDev DDT!)

GSBug lets you do that with an IRQ (Interrupt Request) instead of a NMI. The only disadvantage of the IRQ is that the interrupt bit must be clear in the processor register for the interrupt to be noticed. The NMI method *always* stops the program. And, if your program is running off into space, interrupting it there probably won't do much good anyway.

So Is It Useful?

I am an avid promoter and user of GSBug. In the months I've had the DDT16, I don't think I ever used it once to debug anything. It's not for a lack of trying, either. It's simply too cumbersome. For native IIGS development, I have to say that GSBug leaves the DDT16 in the dirt. (That's not to say that the DDT16 doesn't have its merits, though, just that GSBug is more usable). I looked long and hard to find a situation where the DDT16 would come in useful, and the only thing I could think of would be for ProDOS 8 applications and stuff where a software debugger simply can't be used (like if you want to break into some of the FTA software or games that use a nonstandard operating systems).

The biggest limitation of the DDT16 is its software. The hardware is marvelously well designed. The NMI button has a fairly long cable that allows me to put it well within reach. The biggest complaint I have with the DDT16 is its step and trace mode. It's very hard to get an idea of what is going on when instructions are separated by 4 lines of register values. But software can be updated. And, the folks at ProDev are always eager to hear new suggestions for their product.

Would I buy It?

No, I don't think so. GSBug, even though it is still a beta product, is the debugger of choice for native IIGS development. Maybe I would get a DDT if I had a IIe, had to debug ProDOS 8 software, or was writing something *very* low-level such as a device driver where speed counted and I didn't care how long it took me to step through the instructions. Otherwise I would recommend that you wait until the DDT16 software is improved. **GS+**

Does Your GS Talk?

It can with the NEW Talking Tools™ Program

Most computers need expensive hardware to talk, but your GS computer will talk with the simple addition of our Talking Tools program. It's easy to use and fun to play with. Talking Tools puts the latest in computer generated speech at your fingertips, giving you hours of added enjoyment from your computer.

Your Computer Can Talk! Talking Tools includes the SpeakIt program, a complete desktop application that will bring your computer to life! SpeakIt is so easy to use, your computer will be able to read words, sentences, and entire documents as soon as you run the software. Easy to use dialogs let you choose a male or female voice, adjust the pitch, change the speed and change the volume. SpeakIt also includes a fully functioning text editor, so you can edit words, phrases, and documents.

Programming Anyone? Adding speech to any of your ORCA programs is a snap using the Talking Tools interface files. As you know, one of the easiest ways to learn to program is by example, and that's the way our manual shows you how to use the speech tools. Even a beginning programmer can follow the step-by-step instructions as the manual steps you through a fully functioning talking program. And because source code can be an invaluable teacher, we've also included the complete source for the SpeakIt desktop application, described above. Source code for both of these programs is provided in C, Pascal and assembly language, so you can use your favorite language.

Get the Most from Your GS! Isn't it time your GS spoke for itself? With Talking Tools you and your GS can enter into this exciting new world today! *System Requirements for a Talking Computer: Apple IIGS computer with one 3.5" disk drive and 1M of memory. Programming with the Talking Tools: ORCA/C, ORCA/PASCAL or the ORCA/M Assembler and 1.125 of memory.* **Only $60**

More Great Programming Tools for Your Apple IIGS

ORCA/C	$150.00	ANSI C with Desktop and Debugger
ORCA Pascal	$150.00	ISO Pascal with Desktop and Debugger
ORCA/M 2.0	$125.00	Macro Assembler for the Apple IIGS
DesignMaster	$95.00	A Visual Programming Design Tool
ORCA/Disassembler	$49.95	Desktop and Text Disassembler
The Ugly Duckling	$49.45	Animated Talking Storybook

1010 1101 0110

The Byte Works Inc.®
4700 Irving Blvd. NW Suite 207
Albuquerque, NM 87114

☎ Or Call (505) 898-8183

If you would like more information about any of our products, please write or call!

GS Memory Board

Retail price: Not available
Typical mail order price: 0 MB - $79
each additional MB - $49

C. V. Technologies
1800 E. Whipp Rd. Suite 200
Kettering, OH 45440
(513) 435-5743

Reviewed by Dean Mitchell

With the release of Apple's HyperCard IIGS, I realized I would not be able to squeak by on my Apple 1 MB memory expansion card any more. Up to this point, I had been reluctant to buy another memory card and throw out my old one, since the old one had cost more (when memory prices were high) than a fully populated 4 MB card costs now! It was just the principle of the thing.

Then came the GS Memory Board to save my pride. This is a memory expansion card that will hold 1 to 4 MB in 1 MB increments. It also includes a piggyback connector to hold another memory board, for a maximum capacity of 8 MB. It is fully DMA (Direct Memory Access) compatible, even with another board piggybacked to it, and will work on either ROM 01 or ROM 03 machines. [Only the first 4 MB of any card, however, can take advantage of DMA. This means that, although the other 4 MB card will work, it will be accessed slowly.] The card also comes with a lifetime warranty.

Caveats

The piggybacked memory board may be only one of two kinds: Apple's 1 MB memory expansion card, or a 4 MB expansion card. Also, in order to use either one, they must be *completely* full; a 4 MB card must contain all 4 MB, Apple's expansion card must contain all 1 MB.

If you want to piggyback a 4 MB card, it must conform to Apple's design guidelines for memory expansion cards—bad news for the many people that own an older Applied Engineering 4 MB card. In the words of the GS Memory Board owner's manual, "The methods used by Applied Engineering to implement their memory products prevent any AE memory card (with the exception of the GS RAM II) from being used in the piggyback connector of the GS Memory Board." If your 4 MB card is not an AE card, I recommend you consult your owner's manual or call C. V. Technologies before you buy to confirm its compatibility with the GS Memory Board. The Chinook 4000 is the only example given in the owner's manual of a 4 MB card that *is* compatible. However, I contacted C. V. Technologies and learned that since the manual was written, the GS Juice+, and On Board RAM cards have been tested and are also compatible. You can also plug a second C. V. Technologies memory board into the first if you remove the second board's piggyback connector.

Installation Is Easy . . . Really!

The GS Memory Board comes with a five-page owner's manual. It is short but very clear. Each possible configuration has an explanation for installation, which went smoothly (though I had to think about it a little—see below), and they offer technical support if you have any problems. C. V. Technologies support is available by phone, or via modem on GEnie, CompuServe, and America Online. The only real problem with the manual is that it does not tell you which way to orient your memory chips (i. e. should the little notch on the RAM chip point up, towards the top of the card, or down, towards the bottom of the card.) However, by looking at the orientation of the notches of the other chips on the card you can easily tell which way the RAM chips should be installed. (On my card, the notches were pointing down.)

Before installation in your IIGS, the GS-Memory Board must be configured. This depends on how much memory you are installing, and what, if any, memory board you are piggybacking. The memory chips you install may need to be put in different banks on the card (there are four banks, each accepting eight 1 MB x 1 chips, the chips must be rated at 120ns or faster), and you may have to shuffle the two jumper blocks included with the board.

So, as an example, if you install 2 MB of chips, you put the 2 MB in banks 1 and 2, and rearrange the jumpers as shown in the manual. Yes, you have to jump through a few hoops, but the manual is very clear and lists the various combinations on charts, and the chip banks and jumper positions are clearly labeled.

Once you have arranged the chips and jumpers in the correct order, you can install your piggybacked card. The slot is affixed to the outboard side of the GS Memory Board, next to the side of your computer. So when the GS Memory Board is installed with a piggyback, the piggyback card is wedged between the two, with the various pins of the piggybacked card against the metal insides of your IIGS. To prevent contact between the two, the GS Memory Board comes with a piece of cardboard (about the thickness of a regular index card) that you must install between the piggybacked card and the side of your IIGS. This is easily enough done, but I would like to see the divider made of something a little heavier. The cardboard can buckle if not installed correctly, and I am wary that the sharp pins on the circuit board will puncture it. Their warning does not make me feel any better: "Turn on the power switch. If nothing happens or you hear a squealing sound from your computer turn the power off *immediately* and re-check that the cardboard divider is correctly installed." Ugh. I closed my eyes when I turned my IIGS on, praying I would not hear that "squealing" sound. All I heard was silence, thank goodness.

Conclusion

The GS Memory Board has worked perfectly for me. It solved my memory shortage so that I did not have to toss my 1 MB card, and it did so fairly cheaply. I am very happy with the purchase and would buy it again. Even if I did not previously own the 1 MB card, I may have bought it for its expansion ability and C. V. Technologies' good reputation. My only complaints are about the thinness of the divider, and its restrictions for the piggyback card. Especially disappointing its incompatibility with AE's older cards which, according to C. V. Technologies, is AE's own fault. **GS+**

Photonix II

Program by the Free Tools Association

Retail price: $45
Typical mail-order price: $40
Not copy protected
Requires 1 MB RAM

ToolBox
6, Rue Henri Barbusse
95100 Argenteuil, France
FAX 0 11 33 1 39 47 44 08

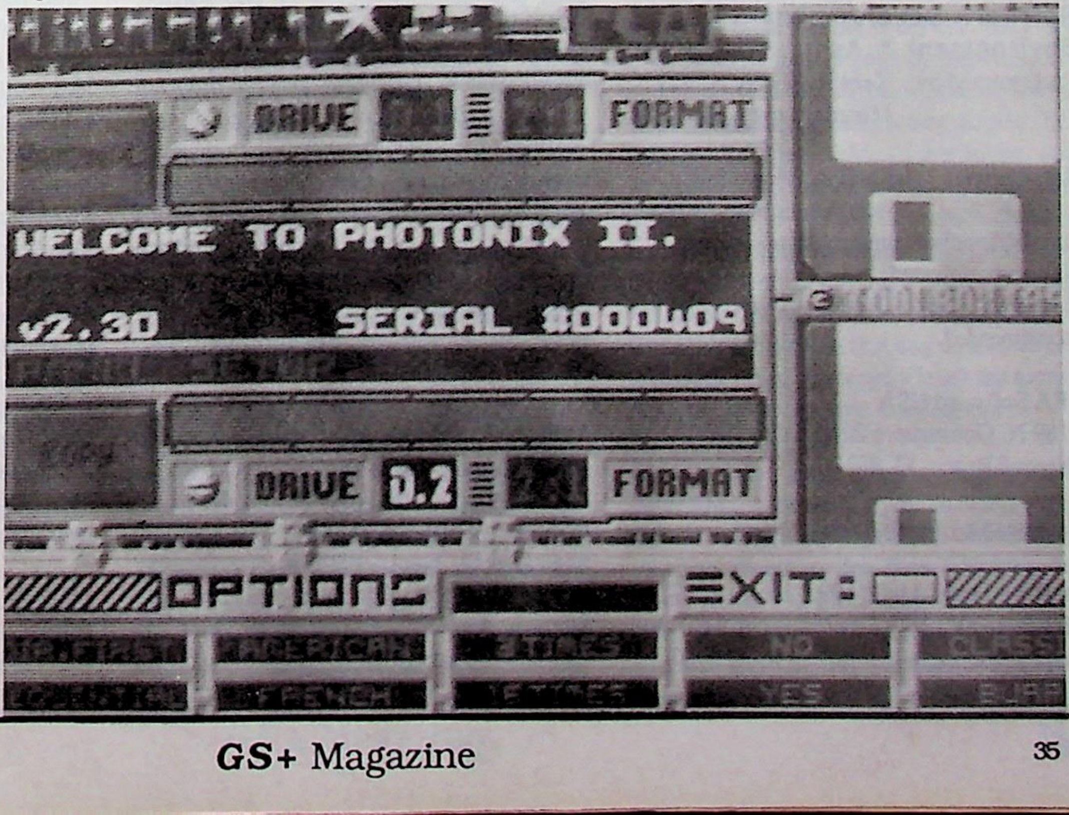
Reviewed by Steven W. Disbrow

The Classic Reborn

If you are a fan of public domain and shareware software, you probably know the name FTA. (FTA stands for "Free Tools Association.") FTA has been responsible for some of the most famous IIGS demonstration programs (Nucleus, Modulae, etc.) and one shareware program that was actually quite useful: Photonix. Photonix is/was a shareware disk copier that allows you to quickly and easily make copies of 3.5-inch disks. Photonix II is a new *commercial* version of Photonix. It is *not* public domain or shareware and contains a few new features to set it apart from its shareware predecessor.

What Can It Do?

Well, first and foremost, Photonix II copies disks. To copy a disk, you just insert it in your first 3.5-inch drive and Photonix II begins to read the disk. A thermometer on screen shows how the read operation is progressing. When Photonix II has finished reading the disk, it spits it out. You then insert the disk you want the copy written to and Photonix II begins to write out the copy. Photonix II does not care if the target disk is formatted or not and it does not care if there was anything on the disk to begin with. Photonix II will simply overwrite everything on the target disk, laying down a new format as it goes along. When the copy has been written, Photonix II spits out the disk and waits for you to insert another. (At this point, I should probably mention that Photonix II can *not* reliably copy disks that are copy-protected. It is intended for use only with disks that are not copy-protected.)


WELCOME TO PHOTONIX II.

v2.30 SERIAL #000409

Most of the other things that Photonix II does, it does to complement this very simple copy procedure. For example, if you need to make more than one copy of a disk (like we often have to do), simply click on the Mass Copy button and Photonix II will write the same copy over and over again until you turn off the Mass Copy option. This makes it extremely easy to copy a mess of magnetic-media in mere moments. (Please forgive me for practicing my alliteration.)

Other Photonix II options let you catalog disks, change the interleave of the target disks, format disks, "optimize" the target disks (by reorganizing the files in memory after the disk is read) and automatically verify copies as they are written. There is even an option to scan your source disk automatically for viruses!

What's New?

Those of you who are familiar with the original Photonix are probably wondering what is new in Photonix II. Well, first of all, Photonix II works with either the ROM 01 or ROM 03 IIGS—the original Photonix only worked on the ROM 01 IIGS.

The main *operational* change in Photonix II is the addition of an Options Panel that allows you to set several operational parameters. From the Options Panel, you can set the display language (English or French), how disks should be reorganized in memory (folders first or sequential organization), the number of times Photonix II should try to read a disk before reporting an error (1, 3 or 16 times), whether or not to "swap" your 3.5-inch disk drives (drive 1 becomes drive 2 and drive 2 becomes drive 1—this is very useful if one of your drives is incompatible with Photonix II) and the sound that Photonix II makes when it ejects disks.

All of that is nice, but the very best change from Photonix to Photonix II is that Photonix II can be installed on, and run from, a hard disk. Just copy the files from the Photonix II disk into a folder on your hard disk. You can then run it just like you would any other program. Well, almost . . . The ability to run Photonix II from a hard drive *should* be cause for celebration. However, it is made almost useless by a very silly oversight: you have to reboot to quit from Photonix II. That's right, you can use the Finder (or another program launcher) to launch

Photonix II, but you can't quit from Photonix II without rebooting. Is it just me or does this seem to defeat the purpose of being able to install it on a hard drive?

Other Problems

Photonix II has a few other problems, almost all of which are caused by the reluctance of the authors to adhere to the rules which Apple has encouraged IIGS developers to adopt. (But then, this is why they call themselves the "Free Tools Association"—they don't like being "forced" to use the toolbox.) First, Photonix II is an 8-bit program. While this is not a problem in itself, it means that you can't get to any of the New Desk Accessories you have on your hard drive that you just ran Photonix II from. If you are thinking, "No problem, I'll just use my Classic Desk Accessories," forget it. Photonix II locks out access to the text-based Control Panel also.

While both of those problems are merely annoying, there are two other problems with Photonix II that are downright frustrating. First, if you have a Video Overlay Card in your IIGS, Photonix II behaves a bit bizarrely. Some graphics will be shifted on the screen and more likely than not, the program will freeze up after only a few minutes. A second, and much nastier problem, is that Photonix II will crash with amazing regularity if you have AppleTalk turned on.

Now, I will admit that most IIGS users don't have a Video Overlay Card and almost none have AppleTalk active but for those of us that do, simply getting Photonix II to run reliably can be much more trouble than it is worth.

Another problem with Photonix II is that, according to the manual, it will only work with the Apple 3.5-inch drive. If you have an Apple 3.5-inch drive and a third party drive (such as the AMR or AE 3.5-inch drives) this can be easily fixed by simply using the Exchange Drives option in the Photonix II Option Panel. If however, you don't have an Apple drive, you are probably out of luck. I say "probably" because I have used the AE 3.5-inch drive with Photonix II and the only problem I had was that the light on the drive constantly flashed. Other than that, the AE drive seems to work perfectly well with Photonix II. The only drive that the manual *specifically* mentions as being incompatible is the AMR 3.5-inch drive.

The final problem that I will mention is that Photonix II saves its parameters (the ones you set in the Options Panel) into your Battery RAM. The original Photonix did this also, but so many people complained about it that FTA released a second shareware version that allowed you to specify whether or not you wanted Photonix to save its parameters in Battery RAM. For some strange reason, Photonix II does not give you a choice, it just goes ahead and writes to your Battery RAM. This is extremely rude behavior for a program and I was very disappointed to see this return in Photonix II. A much better solution would be for Photonix II to save its parameters in a configuration file on disk.

Conclusions

Looking back over this review, I'm afraid that I've skipped one very important point: Photonix II is an *excellent* disk copier! It is fast, reliable (if you don't have a Video Overlay Card and don't have AppleTalk turned on) and it makes great copies. We have used it to copy most of the disks for the last two issues of *GS+* and the only bad disks we have had have been the ones that the Post Office destroyed.

However, Photonix II can be extremely frustrating to use. The multitude of problems I mentioned above, not the least of which is the inability to quit without rebooting, are so frustrating that I am reluctant to recommend the program to anyone but FTA fans and people that have to duplicate lots of IIGS disks. **GS+**

Software Development Environment & Assembler

Programmed by: Kees Buijs and
Mannie Garmy

Retail price: Shell & Editor: $99
with Assembler: $129
Available only from publisher
Not copy protected
Requires 1 MB RAM, a hard disk is recommended.

SEA Software USA
1189 N. Oceanshore Blvd. Suite 189
Ormond Beach, FL 32176

Reviewed by Jonah Stich

In the March-April 1991 issue of *GS+* Magazine (V2.N4), I raved about the value to programmers of the new MAX/Edit editor. Imagine, then, how excited I was to hear of a completely new shell *and* assembler from the authors of MAX/Edit. Unfortunately, this is not an easy program to get. SEA Software is based in the Netherlands, and although they have a branch in America, none of the mailings I received had a phone number (I don't remember the last time I ordered a program by mail). The Software Development Environment (SDE) is, however, also being sold by Complete Technology, Inc. (phone: (303) 758-0920).

SDE was, for the most part, worth the trouble to get. I was hoping for a powerful shell that would bring APW into the world of System Software v5.0, and that's almost exactly what I got. SDE is an extremely powerful shell with more utilities and features than any competing product. It also follows almost all standard conventions for shells, so if you've used another shell before, you should be able to dive right in and get to work. For those who would like a little help, SDE comes with almost five hundred pages of documentation. It is well laid out and covers almost every aspect of the program. Unfortunately, there is no index, so it can take a bit of time to look through the 500 pages for the information you want. Also, the documentation was written by the programmers (who are not native English speakers), and although it is better than some documentation written by American programmers, it does lead to some interesting phrases. Very rarely, however, does this cloud the meaning of the instructions.

SDE is a command-line-based shell, similar to APW/ORCA and the Merlin and ProSel shells. Like those shells, it has a few built-in commands, and is designed to be extended by external utility programs. The command line editing in SDE is better than any other shell I have used—command line entry can be toggled between insert and overstrike mode, and it supports delete left and right, go to beginning and end of line, move to next and previous word, insert space, clear until end of line, and even special keys that move the cursor to the next or previous folder in a pathname (folders must be delimited by slashes, not colons). There is also support for the Apple Extended Keyboard, adding commands to cut, copy, paste, get help, invoke the editor, display a catalog listing, change the current prefix, and more.

The SDE shell is fast, powerful, and well integrated. Much of this is managed through the use of a custom tool set that the shell, included utilities, and editor all make use of. Unfortunately, this tool is a system tool set, which means that SEA Software had to "steal" a system tool set number. At some point in the future, it is conceivable that Apple will release a new tool set that will have this tool set number, at which time the SDE shell will break. More important however, at least for the time being, is that the tool set must be on your system disk for the shell to operate. This means that if, during the development of a program, you are forced to use an alternate system disk, you will not be able to run the shell. If a user tool set had been used instead, both of these problems would have been avoided. Hopefully SEA Software will see the light and release a new version that utilizes a user tool set instead.

The tool set is, however, extremely powerful (128 pages of the documentation are dedicated to it alone) and the reliance of all parts of the SDE environment upon it leads to great consistency and integration. For instance, the tool set is in charge of macro recording, and it lets you do it everywhere. Macros can be used in the shell, editor, and all of the utilities. A macro can be designed that runs three utilities and then start the editor, or one that saves your source in the editor, exits to the shell, then compiles the program you were working on—all at the touch of a key.

The editor is also well integrated into the shell—you can enter it by merely typing "e", and can drop back out to the shell by pressing escape. The editor is actually a subset of the MAX/Edit editor. Unfortunately, many of the most useful features were removed, such as the auto loader and support for multiple tab setups. Also missing is the ability to specify files to be loaded on the command line. You cannot type "edit myfile," you have to enter the editor and then specify which files to load. The editing commands are all still as powerful as those of MAX/Edit, meaning that the editor is one of the most powerful text editors I have seen for programmers, but some of the features that were removed were replaced by substandard alternatives. Most notable among these is the tab ruler support—in MAX/Edit there is a special utility that allows you to edit, load, and save multiple tab rulers; in SDE, you have to edit and then recompile an assembly source file to customize the tab rulers. And, worst of all, you cannot substitute MAX/Edit for the built-in editor—MAX/Edit seems to get confused by SDE's tool set (which is the same stolen tool set number as MAX/Edit's tool set).

The shell itself does all of the things that are standard in other shells, most notably the APW/ORCA shell. Usually, however, it does them in a more powerful way. It supports output redirection, command history, and an extensible command table (i.e. you can add your own utilities). It also has a myriad of advanced features, such as prompt specification, preservation of the history list around S16 launches, command shortcuts, and an excellent help utility. The prompt in SDE is user definable—it can be a "#" as it is in APW/ORCA, or the string "Feed me-->", if you'd like. It can even be the current prefix, meaning that you only need look at the prompt to see what folder you're in.

All commands in SDE can have shortcuts. When you make an entry in the Command Table (a file that SDE looks through to learn all of the command names), you can use uppercase letters in the name to specify a shortcut. For instance, the edit command is entered in the table as "Edit," meaning that you only have to type "E" to start the editor. This Command Table is also reloadable from the shell, meaning that you need not quit and restart the shell to install a new utility.

SDE comes with nearly sixty utilities for file, disk, and program maintenance. All of the utilities respect uppercase and lowercase in filenames—if you create a folder **MyFiles**, it actually names it **MyFiles**, not **MYFILES**. The utilities also all understand colons as path separators, in addition to the standard slashes. SDE allows you to define up to eight utility directories that it will look in to find the utility you would like to execute. Also, parameter "switches" are standardized across all of the utilities. All of the utilities that deal with files use a +H switch to make the command apply to hidden files and a -H switch to ignore hidden files. Similarly, the manual claims that all of the disk utilities such as **Format** and **Erase** use a +H switch to "force use of HFS FST"—you can decide what this means for yourself.

Many of the utilities included with SDE are duplicates of those provided with the APW/ORCA shell, usually with increased functionality. For instance, the **Filetype** command allows you to set both file type and auxilliary type, where as APW/ORCA's **Filetype** command only allows filetype changing. The **Catalog** command supports a compressed format, in which filenames are shown in multiple columns instead of the usual one entry per line. It also supports page by page display, alphabetical sorting of the filenames before display, and recursive directory display (i.e. catalog all folders inside of this folder, and all folders inside of them, and so on). There is also an **Info** utility which displays a catalog listing in which the file type is identified by a descriptive text string (example: "New Desk Accessory"). Unfortunately, there is a bug in this utility that causes it to improperly label the access bits (read enabled, write enabled, etc.).

Also included are a group of utilities that APW/ORCA doesn't have. Among these are a **FindFile** command that searches the specified directory for files matching the given filename, a utility to backup files that have changed since the last time the program was run (very similar to ProSel's incremental backup command), a utility that lets you power down and/or restart the IIGS right from the shell, a utility that quickly sorts text files, one to display info on loaded tool sets, another to identify tool error codes, and a **Parm** utility that lets you set all of the parameters in the IIGS's Battery RAM (those that you would normally set from the Control Panel), as well as about three dozen shell-specific parameters. This last utility is particularly nice, as it can be called from a shell script file (a file that contains a list of commands for the shell to execute), allowing you to create scripts that change one of the shell's parameters, execute a utility, then restore the previous setting of the parameter.

Unfortunately, SDE does not recognize APW/ORCA script files. You can, however, change the type of the file to an SDE script and it will execute perfectly. In addition to normal scripts, however, SDE also has a special Menu script file used by the **Menu** utility. This incredibly powerful utility allows you to define custom text menus, complete with special header, footer, and "plain text" items that can't be selected, margins, custom frames, checkmarks, and keyboard equivalents. In response to user selections from the menu, submenus can be opened, menu variables can be set, and programs and utilities can be launched. If you prefer the menu-based interaction of ProSel and Merlin to the command line of APW/ORCA, you can use SDE's **Menu** command to create custom menus that save you from ever needing to see the command line.

The people who wrote the SDE shell and utilities also wrote a fast assembler to go along with it (the shell is available with or without the assembler). While the shell is extremely similar to the APW/ORCA shell, the assembler is extremely similar to the Merlin-16+ assembler. It uses a syntax which strongly resembles that of Merlin, and its output (and speed: it's about twice as fast as the **ASM65816** assembler) while assembling are quite like Merlin's. Unfortunately, this leads to a bit of a problem, as most people who are used to command line interaction with a shell are used to the APW/ORCA assembler format, and those that like the Merlin source format aren't used to a command-line-oriented shell. The assembler seems to be fairly robust, and it has a huge number of features and options (it even supposedly recognizes 65832 opcodes). It doesn't, however, split its assembling and linking into two separate functions. For those who use the SDE assembler exclusively, this is not a problem. For those who would like to link together code compiled with the SDE assembler and that created by another assembler or compiler, well, you can't. Also, the SDE assembler is not a standard shell language, so you cannot install it on your APW/ORCA shell.

There are, unfortunately, some particularly crippling drawbacks to the SDE shell. Chief among these is its support of APW utilities. The programmers at SEA software decided for some reason that the standard calling conventions for shell utilities weren't good enough and created their own protocol for passing parameters to utilities. Cleverly realizing that many users would be coming from APW/ORCA and would like to use their old utilities and languages, the programmers created an "**APW**" utility that passes parameters to utilities in the format used by all other shells. Well, it almost passes the parameters in the format used by other shells. This **APW** command works well with most utilities, but it has a few drawbacks. Most important of these is that it doesn't recognize multiple filename arguments. For instance, I was working on a large, multi-segment program. I compiled each segment separately, then used the command line "Link o/File1 o/File2 o/File3 o/File4 keep=o/Program" to create the final program. This command line works fine in APW/ORCA, but the **APW** utility only understands one filename argument, so it converts this command into "Link o/File1 keep=o/Program". Furthermore, some APW/ORCA commands are internal to the shell, and there are no duplicates of them in SDE (**ASML** is an example). SDE tries to finesse this problem by remembering the filename typed after an **ASML** command and passing it to both **compile** and **link**. Usually, however, this results in an error at the link stage when SDE tries to use a command line such as "link myfile.asm." This can be avoided by entering the proper compile and link commands yourself, but it does take an extra step, and programmers are not known for their extraordinary patience.

Also annoying (though not nearly as prohibitive) is the fact that SDE uses its own text output routines, and it doesn't seem to initialize them properly when an APW/ORCA style utility is used, so that the first line of output from the utility is not displayed. Furthermore, SDE doesn't seem to allow aborting utilities by pressing Command-period. Another little feature that can lead to problems is program launching. To launch a shell utility (EXE) or ProDOS 8 (SYS) file, you merely type its name on the command line (or use the **APW** utility, if it's an APW/ORCA style utility). To launch GS/OS (S16) files, however, you must quit to them, as in "**Quit AppleWorksGS**." And finally, AppleSoft BASIC files must be run using the **BASIC** command (i.e. "**BASIC MyProg.BAS**"). Remembering which form to use can be hard, and it's not made easier by the fact that if you forget to **Quit** to a S16 file, SDE will run the program as an EXE file, usually causing the system to crash. (SDE does redeem itself for BASIC users, though, as it has two utilities that allow you to convert AppleSoft to text and text to AppleSoft. This allows you to convert your BASIC listings to text, edit them in SDE's powerful text editor, then convert them back to AppleSoft files and run them.)

The Software Development Environment, in and of itself, is a wonderful program which has very few bugs and is much more powerful than any other similar product available for the IIGS. It does, however, have some major compatibility problems that make it a bit tricky to upgrade from other shells. Hopefully SEA Software will release an upgrade that fixes the problems with the **APW** command, makes the tool set a user tool set, and adds some of the functionality that was removed from the editor. When this happens, SDE will be, without a doubt, the best shell for the IIGS programmer. As it stands now, it is still a wonderful tool to those for whom APW/ORCA compatibility is not a major concern, but to those of us with hundreds of kilobytes of source code for APW/ORCA languages, SDE has too many problems executing APW/ORCA style commands to make it viable for major program development. **GS+**

Jungle Safari

Program by Dlyan Gladstone

Retail price: $49 ($59 with backup disk)
Typical mail-order price: $38
Not copy protected
Requires 1 MB RAM

Orange Cherry Software
Box 390 Westchester Ave.
Pound Ridge, New York 10576-0390
(800) 672-6002
(914) 764-4104

Reviewed by Greg Zimmerman

Jungle Safari is a newly released Apple IIGS specific children's educational software offering from Orange Cherry Software. It is one of the latest programs in a long line of IIGS-specific software published by Orange Cherry.

Jungle Safari features a simplified user interface, through which the child navigates four different modules containing scenes typical to Africa. Each module contains different plant and animal life shown in high-quality color graphic scenes. Each scene allows the child to click using the mouse to learn about any plant or animal shown in the program. Jungle Safari comes on two 3.5-inch disks, is not copy protected, is hard drive installable, runs on either a ROM 01 or ROM 03 IIGS, and is System Software v5.0.4 compatible.

The program is accompanied by the now familiar (at least to me and my kids) Orange Cherry manual, which accurately and simply describes the operation of the program, as well as offering hard drive installation instructions, and suggested activities away from the computer for the program material.

What It Does

The Jungle Safari title screen allows the child to select from among six options. Four of the options are the modules that take the child either for a trip down a River, into the Plains, into Tall Grass, or into the Rain Forest. The fifth choice is to print one of five pre-made groups of animal pictures, and the sixth selection allows the child to quit the program.

Three of the learning modules are very similar both in looks and operation. In Tall Grass, Rain Forest, and River Tour, various screens of animal and plant life indigenous to the particular chosen setting are presented on screen, and the child may use the mouse to click on just about anything in the screen graphic to bring up more information on the chosen picture. An information box then fills with text containing interesting facts about the plant or animal chosen by the child. The child controls the pace at which the screens change so there is plenty of time to explore each of the items presented in each graphic.

In the fourth module, Plains, the child is at the wheel of a jeep, trying to locate all the animals in the area. The child must maneuver the jeep by clicking on arrows on the steering wheel, and only when the jeep is in the proper position facing the particular animal, may the child find out information relative to the newfound quarry. In this module, the child may click on a box to hear the animal's name, click on another box to see what the animal's baby would look like, and click on a third box to bring up several sentences of facts concerning the animal.

Examples of the types of plants and animals found in the four learning modules span approximately 80 possibilities including snakes, chimps, termites, leopards, vultures, etc.

The printing module allows the user to print one of five pictures containing four different examples of African wildlife. Printing is only in black and white, and unlike the rest of the program, is done from a desktop environment which closely resembles the one specified in the Apple Human Interface Guidelines.

The Good News

Jungle Safari is a pretty entertaining program for young children. The graphics are great, and the manner in which the material is searched out by the child is excellent. Also, the learning is self-paced, which alleviates pressure on the child, and there are sufficient different kinds of animals and environments to explore that there is no problem keeping a child at the computer using this program for 30 to 60 minutes. Also, because the subject of different and sometimes unusual animals is generally interesting to children of all ages, they may use the program again and again. Finally, because of the way in which the program interface is laid out, the written words that describe facts concerning the chosen animal and plant are not necessary to entertain and interest a young child. So the program can be used by children in a wide range of age groups, my best guess being age 3 (my little beast got to see some of his *really* close relatives) to age 12, and possibly even age ranges up into and including high school.

This program takes advantage of both the best graphics capabilities of the IIGS and the point and click ease of use found in the best children's programs.

The Bad News

As with most Orange Cherry software, loading from floppy disks takes quite a while. On a stock ROM 01 IIGS, booting from the 3.5-inch disks, it took over one minute to the main menu screen, and then anywhere from 25 to 55 more seconds to get to an activity depending on which module is selected. Of course, hard drive boot times are a little faster. Booting from a 100 MB Vulcan from the System Software v5.0.4 Finder on a TransWarped IIGS, the loading time to the main menu was reduced to 12 seconds.

One other minor complaint is the fact that when you quit the printing module, you also quit the program. You are not returned to the main menu.

Do I Recommend It?

Jungle Safari is one of the best Orange Cherry programs on the market. I have reviewed lots of their software, and have had a hard time recommending that anyone buy most of it. However, (drum roll please) Jungle Safari is worth the purchase price. It is entertaining, the subject matter is interesting, and the graphics are excellent. The fact that it can be used with a wide range of ages is a big plus, and even though it is not a textbook on African plant and animal life, my kids learned from it, and were happy to do so. GS+

Space Shuttle Word Problems

Program by Richard E. Dye

Retail price: $49 ($59 with backup disk)
Typical mail-order price: $38
Not copy protected
Requires 768K RAM

Orange Cherry Software
Box 390 Westchester Ave.
Pound Ridge, New York 10576-0390
(800) 672-6002
(914) 764-4104

Reviewed by Greg Zimmerman

Space Shuttle Word Problems is a children's educational math and reading program published by Orange Cherry Software. Orange Cherry has continued to produce new software for the Apple IIGS throughout the last two years, so Space Shuttle Word Problems is one of the latest in what is becoming a rather large collection of IIGS-specific educational software put out by Orange Cherry.

Space Shuttle Word Problems uses the Space Shuttle (what was your first clue?) as a topic with which to cover a series of 30 math problems in what Orange Cherry calls "reading challenges." This means that all the math problems are presented as word problems, and they all share the common topic of a Space Shuttle mission.

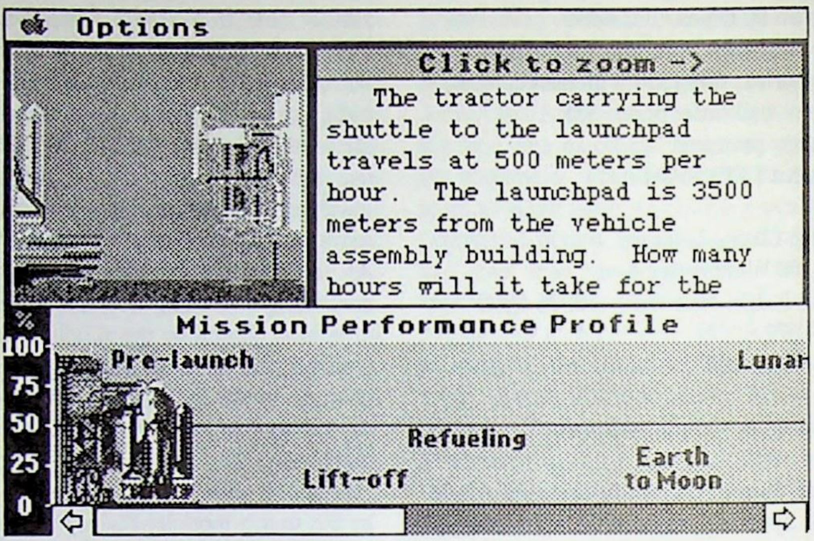
The Nitty Gritty

Space Shuttle Word Problems comes on one 3.5-inch disk and is not copy protected. It is compatible with System Software v5.0.4, is installable on a hard drive, and will run properly on either a ROM 01 or ROM 03 Apple IIGS.

The program is accompanied by a short but well-written and easy-to-follow manual describing the program's functions, detailing hard drive installation instructions, and suggesting activities to reinforce the program material away from the computer.

How Does It Work?

Space Shuttle Word Problems takes the user through a series of 30 math problems which all relate to a Space Shuttle mission. The word problems appear written out on the screen. When the user is ready to input the (hopefully) correct answer, the user must click on a small calculator icon, which brings up a large calculator. The calculator is only used to input answers, it will not allow the user to make computations. If the user enters the correct answer, the program moves on to the next problem. However, if the wrong answer is input, the user is given another chance to enter the answer correctly, after which the program tells the user the correct answer, and offers a detailed, step by step explanation of how the correct answer was computed.



The tractor carrying the shuttle to the launchpad travels at 500 meters per hour. The launchpad is 3500 meters from the vehicle assembly building. How many hours will it take for the

The problems are broken down into ten sets of three "challenges." After each group of three problems, the user is shown a small but high quality animated screen graphic depicting the completion of that particular phase of the shuttle mission, and then the program goes to the next phase, or the next set of three word problems. This continues until all 10 sets of problems are answered, and the shuttle touches down for a landing after the completion (successful or otherwise) of its mission.

The numerics in the problems are changed so that the numbers and answers are not the same each time the program is used. Also, the program has a window containing a graphic chart at the bottom of the screen detailing the phases of a shuttle mission, and allowing the user to chart her progress as she works her (Using the feminine for the anonymous generic sounds strange, doesn't it?) way through the various phases of the Space Shuttle mission.

At the end of the mission, the user is told what percentage of problems were answered correctly, and given the choice of going on another mission, or quitting the program.

Space Shuttle Word Problems pretty much follows the Apple Human Interface Guidelines. There is a menu bar present at all times the program is running, New Desk Accessories are supported, and there is an instruction screen selectable at almost all times no matter where in the program the user runs into a problem.

Orange Cherry recommends Space Shuttle Word Problems for ages nine to eleven. While some of the material is suitable for ages down as low as six or seven, keep in mind that the user must know how to read, or this program will be a waste of time.

Is The Program Any Good?

Well, it is kind of an interesting approach to math on the Apple IIGS. Orange Cherry has selected a topic that almost all kids will find interesting, and then packaged math problems in word format around the interesting topic.

Combining this with the very high quality of the graphics and animations, Space Shuttle Word Problems can keep a child's interest level up through all 30 problems.

It's colorful, interesting, and gets away from the old drill and practice method of math teaching. It also helps with reading, and helps the child to learn to logically sort out the information given in the word problem so that the necessary information can be separated from the clutter and then arranged in such a way as to lead to a solution to the problem at hand.

The always available instruction screen is a good idea, and changing the numbers in the problems also extends the life of the program.

Problems?

Space Shuttle Word Problems follows in a long line of software that is slow to load from floppy disks, and it is no exception. On a stock ROM 01 IIGS, booting from the 3.5-inch disk, the loading time, including two required mouse selections to keep the loading moving forward to the actual program, was approximately one minute and thirty seconds (1:30). But as usual, this problem was solved when the loading was done from a 100 MB Vulcan, on a ROM 01 IIGS equipped with a TransWarp. Launching from the System Software v5.04 Finder, the time to load was reduced to 14 seconds.

Space Shuttle Word Problems also has what I regard as some minor design flaws. For instance, the user cannot go back to redo problems that have already been passed over or done once. The "mission" will only move forward. Another example is the window that contains the word problem on screen. It has the little box in the upper right hand corner to shrink (or expand) the window so that the mission progress chart at the bottom of the screen can be uncovered (when fully sized, the text window blocks part of the chart). However, the window does not have a scroll bar, so to read the entire problem the user has to "unshrink" the window and have the progress bar partially hidden. This is more of a slight nuisance than anything else, but the addition of a scroll bar in the window would seem to have been easily accomplished. One other programming "problem" is that the calculator only accepts answer input, and is not usable for figuring out the answers. In other words, it's not a calculator at all.

The only other bad news about the software is that with only 30 problems to do, the program will not exactly have the same half-life around the house as the radioactive debris currently polluting more than just a few square miles in the Soviet Union. Orange Cherry tried to eliminate this as a problem by changing the numbers in the "challenges," but it still is a problem for the software's longevity with any individual user. Because the basic format of the program is a good one, this problem would best be solved through the offering of additional data modules with significantly different content than the problems contained in the program disk. Or, just as many of the Orange Cherry offerings have two disks, a second disk could have been added right from the start, featuring more word problem selections.

The Wrap-Up

Ah, the wrap-up. I have never had a problem with "The Wrap-up" except when doing reviews of Orange Cherry software. Space Shuttle Word Problems is no exception. It's good, but I wish there were more to it for the money. Will it teach your kids math? *No.* Will it help reinforce math concepts in an interesting and fun format? *Yes.* Do you have enough kids to justify spending the money on a good but somewhat shallow program? Of course, if you have a lot of kids can you afford to spend the money on any program?

I have four kids, and if I didn't have this review copy provided free by the publisher, I would still buy the software because it is a good program, and all the kids would get some use out of it at varying times. And it is a good educational tool to add to the arsenal that keeps them in front of the computer instead of in front of the idiot box.

Tough call on this one, but it's for the kids, so what the heck. They are a colossal money pit already, so what's another $38? Buy the program. **GS+**

KANSASFEST BABE DATA SHEET

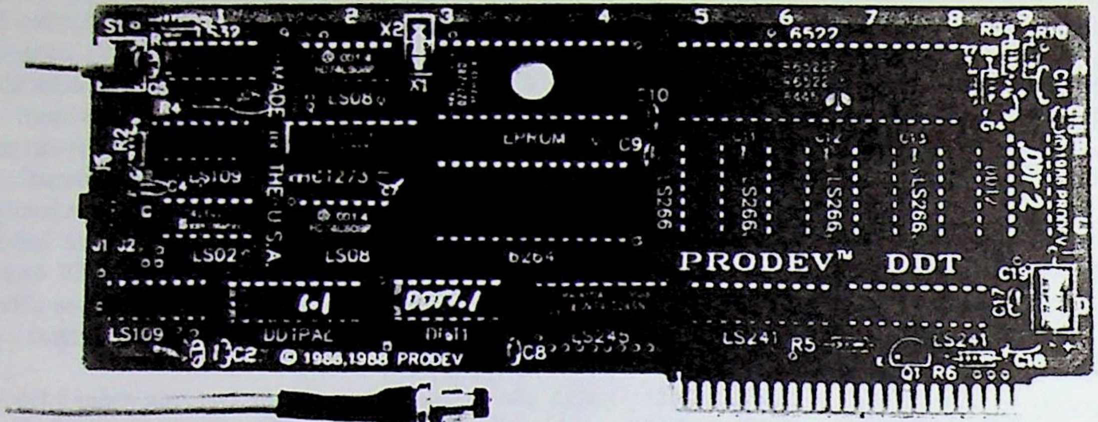
Name: Patty Westerfield

Ambition: To open a seafood restaurant in the deserts of New Mexico.

Turn-Ons: Big inflatable whales. Guys who write compilers.

Turn-Offs: The phrase, "Just one more bug to find, and I'll be right home!" People that make things up and attribute them to me just to fill a few stinking columns in their stupid magazine.

The most powerful program debugging tool available for the Apple II[‡] computer!



The **Prodev**™ **D**evelopment & **D**ebugging **T**ool

Just look at some of these features:

CONTAINS A HARDWARE BASED STOP-ON-ADDRESS.
This feature allows you to stop your program when the address you specify is referenced for any reason.

THE BUTTON-INTERRUPT
Regain control of programs that are lost or pause the operation of a program.

SET REAL-TIME RANGES
During tracing if a JSR instruction places the program in a real-time range the program will start real-time execution until the instruction following the JSR in memory is reached.

HALT YOUR PROGRAM WITH 1 OF 50 BREAK-POINTS.

GIVES YOU ACCESS TO THE PROCESSOR'S REGISTERS.

FULL SUPPORT FOR THE 6502, 65C02, 65802* AND 65816*.

* - IIGS version only

TIME THE OPERATION OF YOUR PROGRAM

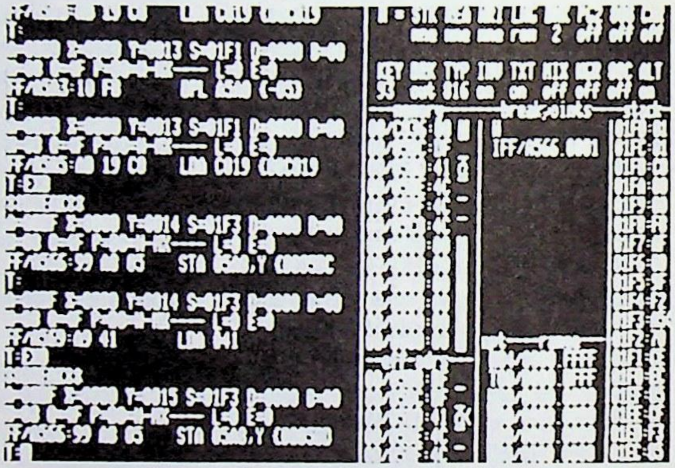
EXAMINE AND CHANGE MEMORY.

SINGLE-STEP AND TRACE FUNCTIONS.

CONTAINS A BUILT IN MINI-ASSEMBLER.

CREATE YOUR OWN TEST ROUTINES

SEND THE OUTPUT FROM THE DDT TO EXTERNAL DISPLAY DEVICES.

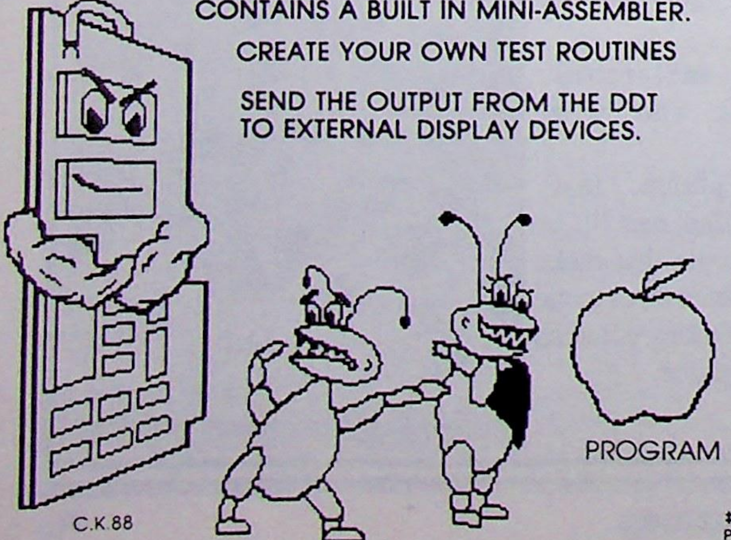


```
[illegible]
T:
[illegible] Y=0013 S=01F1 [illegible]
FF/[illegible]:10 F8        BPL [illegible]
T:
[illegible] Y=0013 S=01F1 [illegible]
FF/[illegible] 19 C0        LDA C019 [illegible]
T:
[illegible] Y=0014 S=01F3 [illegible]
FF/[illegible]:99 [illegible] 05  STA [illegible],Y [illegible]
T:
[illegible] Y=0014 S=01F3 [illegible]
FF/[illegible] 41           LDA [illegible]41
T:
[illegible] Y=0015 S=01F3 [illegible]
FF/[illegible]:99 [illegible] 05  STA [illegible],Y [illegible]
T:
```

```
KEY [illegible] TYP INV TXT [illegible] [illegible] [illegible] ALT
S3  [illegible] 816 on  on  off off off on
--------break-points--------
IFF/[illegible].0001
```

WINDOWED DISPLAY SCREEN
The output from the Prodev DDT is provided in an easy to understand windowed format.

SET PROTECTION RANGES
Alerts you if your program ventures outside the given code range or inside the given protection range.



SYSTEM REQUIREMENTS

The Prodev DDT is available in two versions.

The Prodev DDT8 requires an Apple //e

The Prodev DDT16 requires an Apple IIGS

All of the above features and more for only $189.00.
To order send check or money order to:

Prodev, Inc.
P.O. Box 162
LaSalle, MI 48145-0162

For telephone orders or for more information

call (313) 241-2786.

$4 Shipping 2nd day air. Michigan residents add 4%

‡ - Apple Apple II, Apple //e and Apple IIGS are registered trademarks of Apple Computer, Inc. Prodev is a trademark of Prodev, Inc.

GeoQuiz

By Xavier Schott, Jean Pierre Curchio, Nathalie Merino, and Luc Serard.

Retail price: $49.95
Typical mail-order price: $25
Not copy protected
Requires 512K RAM
(1 MB recommended)

PC Globe, Inc.
4700 South McClintock
Tempe, Arizona 85282
(602) 730-9000

Reviewed by Dave Adams

PC Globe has won rave reviews for their geography software on the PC platform. GeoQuiz is their first program for the Apple IIGS. GeoQuiz is a geography game that focuses on locating countries and capital cities. It features user-definable variations in the quizzes that keep the challenge from wearing out too quickly. Games can be saved and replayed. GeoQuiz comes on one 3.5-inch disk and is not copy protected. The game comes with a detailed manual that explains all facets of game play (including scoring) quite well. It works fine on a hard drive under System Software v5.0.4 and supports desk accessories.

The Good Stuff

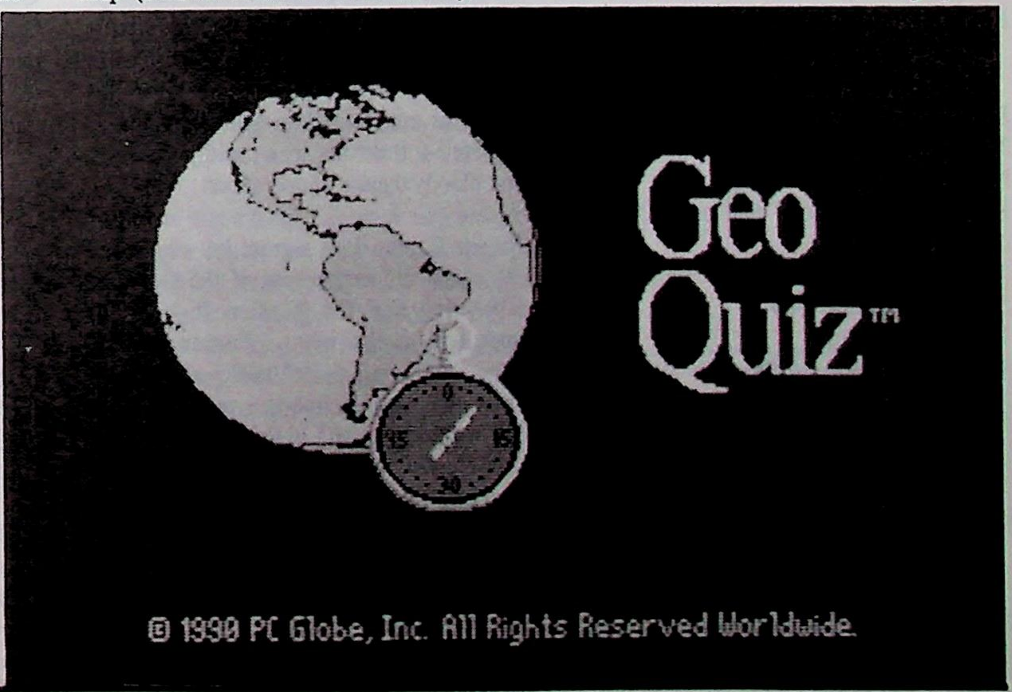
GeoQuiz is exactly what the name implies—a geography quiz. I used this program with some of my Geography students and their knowledge of locating these subjects was measurably improved. The object of the game is to guess the correct answer in the shortest amount of time. One of the best features of the game is that you can make it more or less challenging. The default game allows you to select the correct country (or capital city) from a list of four choices below a map of the country's region. The country (or city) flashes to identify itself. You simply click on the correct answer. If you select the wrong answer the answer that you clicked on dims and you must click on one of the remaining three choices. Clicking on the wrong answer has an adverse effect on your score. Clicking on the correct answer scores a number of points based upon the time that has elapsed. Taking a long time scores fewer points than a quick response. You can also just throw in the towel and get the correct answer by forfeiting all of your points for that question. Now that basically is nothing more than a computerized multiple choice quiz with a "Beat the Clock" twist. Once you have mastered that you can increase the difficulty by changing the type of quiz from "Name It" (Where the capital is provided as a hint) to "Recognize It" (No capital hint) or the dreaded "Locate It" (You have to click on the right location).

Other features of the game can be changed. You can require the answers to be typed in at the keyboard rather than clicking with the mouse. This allows many more points to be scored as the answer must be correctly spelled or it won't count. This can be a problem when you have to type in "Ouagadougou" or "Antananarivo" (the capitals of Burkina Fasa and Madagascar, respectively) and you are trying to beat the clock. Quizzes can be any number of questions from 1 to 170. You can provide for a number of hints from 0 to 3. Using hints deducts points from the score. When using the "Name It" game you can even change the map cues to no map (just a generic world map with no flashes), region map (smaller section of the world), or flash country (obviously this flashes the country). At the conclusion of each quiz a scoreboard shows you your correct first choice responses, number of missed questions, score, and the percentage of correct answers. Once you have the countries of the world down pat you can start naming the capitals. "Beat the Clock" was never so flexible.

In addition, GeoQuiz has an Atlas/Database for each country in the world. Each country has selected facts from the following categories: Region and World Map views, Capital city and location, Independence Day, GNP (Gross National Product) per capita, Area, Population, Flag, Religions, Currency, UN Date, National Holiday, and Languages. Selecting a fact brings a colorful picture to the screen to illustrate the fact. The Atlas can find countries and capitals (Where is Burkina Fasa anyway?) and can be accessed during the game. Of course accessing the Atlas results in a huge penalty to your score for that question. Hints for the game are randomly selected from the Atlas Database. Accessing hints also deducts points from the question's score. Another nice feature is the ability to save screens as Apple Graphics formats within the program.

But . . .

There are a number of improvements that could be made. First of all, the graphics



© 1990 PC Globe, Inc. All Rights Reserved Worldwide.

are distressingly similar to IBM EGA type pictures. The game does not really take advantage of the IIGS's ability to create exceptional graphics. Considering that the game operates in 320 mode, this is hard to understand. It also was written in ProDOS 16. I would much rather see a two-disk GS/OS program than a one-disk ProDOS 16 program—but in fairness that is a personal opinion. The boot time is not that bad for ProDOS 16—around 1.5 to 2 minutes. Booting from a hard drive with an accelerator takes less than 10 seconds.

The hints that GeoQuiz gives can be quite inane at times. Since they are randomly chosen they can range from the obvious (Currency = Iraqi dinar) to the obscure (UN Date = 1945. Gee, weren't there only about 75 countries that joined the U.N. in 1945? I mean, that's when they founded it, right?). Trying to figure out which country you are looking at with your one hint being "Population = 9000" can be really frustrating. The hints are not usually useful as hints. Knowing the population or area of a country is ridiculous without having something to put it into context with. For example, most people can not tell you which country has an area of 8,649,498 square miles. But, if you tell them it is the largest country in the world they might be able to guess that it is the Soviet Union. In other words the facts by themselves are not good hints. Hints should consistently be able to lead you to the answer, not confuse you even more. Something needs to be done for selecting small countries (particularly small island nations) to make it less difficult. Hitting some of those with the tip of the mouse arrow requires greater precision aim than Sergeant York had.

An improvement that I would like to see is the ability to create custom quizzes. As it now stands you can set the number of questions but the countries are chosen at random. I teach Geography by examining regions of the world and would appreciate being able to have the students focus on Europe or Asia rather than having a grab bag approach to the quiz. As it now stands I get what they give me and I have to hope that it covers what I would like. You can save questions sets but there is always a mixture of countries from several regions. Of course, it could be that PC Globe is trying to remind us that we all inhabit the same planet and are a global community.

Worlds Apart

Some people may compare this program to MECC's World GeoGraph. *Don't!* They are two entirely different types of programs. GeoQuiz is a drill and practice quiz type game. Although it does contain an Atlas/Database, it requires a lot more work to make meaningful interpretations than World GeoGraph. World GeoGraph has superior graphics, depth, information, and comparative abilities, yet it cannot match GeoQuiz in teaching locations of countries and capitals. If you want to find them, use GeoQuiz. If you want to compare everything about them and understand them use, World GeoGraph. They are both outstanding programs from excellent companies.

So Does It Work?

Yes. GeoQuiz is an excellent quiz program. It has strong expandability and can increase the challenge as you increase your knowledge. It is as up to date (there is only one Germany) as can be expected. It will definitely improve your location skills for capitals and countries. Kids take to the game quite easily and enjoy seeing their own progress and improved scores. They can also work at their own pace. It can be used to sharpen understanding and reasoning by writing down facts from the database and comparing and interpreting them. Although it is primarily a drill and practice type exercise it does get results. It is a great value for the price. I highly recommend GeoQuiz. **GS+**

Panzer Battles

By Roger Keating, Danny Stevens, and Ian Trout

Retail price: $45
Typical mail-order price: $29
Not copy protected
Requires 1 MB RAM and one 3.5-inch drive

Strategic Studies Group
1747 Orleans Court
Walnut Creek, CA 94598
(415) 932-3019

Reviewed by Dave Adams

Panzer Battles is the second Battlefront Gaming System release by SSG. (See *GS+* V2.N4 for a review of Halls of Montezuma). It is very similar to its predecessor Halls of Montezuma (HOM). As a matter of fact, it can be said that they are just alike, only different. Panzer Battles recreates the Russian Front tank battles of World War II. Both the Germans and Russians fielded immense armored formations in a rapidly flowing and bloody theatre of operations.

Panzer Battles does a great job capturing the clash and maneuvers of the German Blitzkrieg and the Russian Juggernaut approach to the war. Panzer Battles comes on two 3.5-inch disks and is not copy protected. It operates under System Software v5.0.2 and higher. Like HOM, it is easily run from a hard drive and supports NDAs. The game comes with the same Game Manual as HOM and includes a new scenario booklet for Panzer Battles. The Game Manual is a booklet that contains an overview of the Battlefront Gaming System and Warplan Construction System. The scenario booklet introduces you to the operations of the game and contains the usual instructions for installing it to a hard drive, etc. The scenario booklet also contains the scenario descriptions for six major armored battles that took place on the Russian Front. For those of you no with knowledge of such matters these were: Minsk, Moscow, Prokhorovka, Kharkov, Kanev, and Korsun.

Missing from the Panzer Battles kit is a fold out map such as the one in HOM. SSG has included the maps in the scenario booklets for easier reference.

Fire And Maneuver

I can't help comparing Panzer Battles with HOM. As I stated earlier, the two games are just alike, only different. They use the same simple to learn gaming interface. I have successfully run HOM scenarios through Panzer Battles and Panzer Battles scenarios through HOM. They seem to have no differences in their systems. The only thing different about the games are the scenarios.

In HOM you recreated the major battles of the United States Marine Corps. With all due respect to the Marine Corps, few of those battles would be considered as examples of maneuver warfare. The Marines have traditionally been the "High-diddle-diddle-straight-up-the-middle-in-your-face" Assault troops. Their Light Infantry organization is not designed for such warfare on a protracted scale. Basically their Infantryman has the same tactical mobility as a Roman Legionnaire. HOM was essentially a slugfest. Stand toe-to-toe with the enemy and hammer at each other until somebody gives.

Panzer Battles is a whole new ballgame. Although it has plenty of bloody assaults it also contains the opportunity for massive breakouts, pursuits, encirclements, and rapid flank attacks. Quite simply, you can flat out move during these battles!

Strategy is much more complex and challenging. It is also a great deal more exciting. Whether you are leading a Panzer Division or a Guards Tank Division you can almost sense the excitement of thundering across the Steppes and racing to your objective. If you liked HOM then you will probably love Panzer Battles.

After Action Review

Panzer Battles has few faults in the actual game play. Most of the areas that could be improved upon are in the Warplan area of the game. Warplan allows you to adapt or create your own scenarios for Panzer Battles.

As mentioned in the HOM review, the Game Manual that explains Warplan could be written in a way to more easily explain some of the construction features. A design tutorial for creating an entirely new scenario would be most helpful. The tutorial for adapting a scenario is missing from the Panzer Battles scenario booklet making it more difficult for someone who does not own HOM to learn how to use Warplan.

A great improvement would be to allow the importation of Super High Resolution (SHR) graphics for maps. As it now stands each hex on the map must be filled in with terrain one at a time. Adding a major river or other dominant terrain feature involves creating custom hexes. A major river with a few curves is a real pain in the neck to create. It would be much more efficient to allow the user to overlay the terrain type onto a SHR graphic without having to fill in each hex.

I still feel that a quick reference card would be nice for beginners. Adding a few more sounds to the game would also make it a bit more interesting. Finally, Panzer Battles has the annoying habit of changing your border color to grey and then forgetting to change it back again.

The Bottom Line

I recommend Panzer Battles for any IIGS owner looking for a finely crafted strategy game. From the opening screen (which features a Soviet T34 tank coming at you with a wonderful shell burst effect while the "Song of the Volga Boatmen" plays in the background) to the fast and furious, combat, this game is well worth the price.

If you liked HOM you can look forward to some challenging and different scenarios. SSG continues to turn out great IIGS products and is definitely the leader in the IIGS war game market. For hours of strategy and fun . . . check out Panzer Battles. You won't be disappointed. **GS+**

Special Note

To the modemless individual who wrote us here at *GS+* Magazine requesting help on locating Halls Of Montezuma scenarios. I have lost your address and have two disks full of scenarios for you! Please write us again and forgive my carelessness. You have not been forgotten!

Dave Adams

Reach For The Stars

By Rodger Keating, Ian Trout,
Danny Stevens and Alan Bell

Retail price: $44.95
Typical mail-order price: $28
Not copy protected
Requires 512K RAM and System Software v4.0 or later

Strategic Studies Group
1747 Orleans Ct.
Walnut Creek, CA 94598

Reviewed by Bill Patterson

Why Another "Galaxy" Game?

Ever since I got rid of my old Apple II and started dragging my IIc around the planet, I've been on the lookout for good multiplayer games. Being stuck in a tent with 40 potentially violent, hung-over GIs is a very strong incentive to come up with some passive method of working off their aggressions, or at least distracting as many of them as possible for as long as possible. Suprisingly, good, original, and *interesting* multiplayer computer games are very hard to find. Computer translations of classic multiplayer board games are the worst. They are usually so dull that whomever you get to play with you probably won't ever come back for another try—if you even finish it at all. Some of the games that I've enjoyed over the years are: Cosmic Balance, Air Combat, Old Ironsides, and Galactic Empires (there were at least three games with this name, I'm referring to the one that was somewhat similar to "Risk" and published by a company called Ursine). Of these old standbys from those days of fear and loathing, only Galactic Empires will run on the IIGS.

Since I'm now living in a dormitory again, (at a space tracking station on a remote desert island) I am again in need of good multiplayer games. Up till now we've been spending our gaming time at Task Force (2 player), Star Saga I (6 player), Mean 18 (4 player) and good old Galactic Empires (20—gasp!—players). But now there's a new folder on my hard drive: Reach for the Stars; The Conquest of the Galaxy (RFTS).

From the first time I read about this program I knew it had all the makings of a classic multiplayer game. Space Wars, Power Struggles, Economic Disasters—and *you* run it, or ruin it, all. This program promised to deliver what previous space war simulations had failed to, a sense of strife and struggle, as well as a feeling of identification with the empire you build. You would have a growing society to care for, instead of just a abstract quantity of points to burn up in the mindless pursuit of more and more armies/ships as in the Risk/Galactic Empire scenarios.

Beginners Impressions

Installation of RFTS is straightforward. Take everything and stuff it in a folder on your hard disk. If you don't have a hard disk, just use any backup program or utility to make a backup before you play. I highly recommend using the hard disk since it speeds things up a lot. Also, the game doesn't come with System Software v5.0 although it seems to work OK with it for the most part (more on that later). As with any game of this complexity, you have got to read the instructions first. This is where I ran into some problems. The game comes with a built-in tutorial (an attempt for which SSG is to be applauded), but for some reason I couldn't seem to follow the tutorial instructions very well. The basic problem isn't really a fault of the game, but the fact that so much information is available at your fingertips. A nit that should be picked at this point has to do with the window titles. Two of the windows you are given at the very beginning (and throughout the game) show the name of the planet but not the name or function of what the window displays. The confusion goes away after working with the game for a little while, and like I said, this game is complex, not overly so, but it's not Pac-Man either. What I would have liked are some illustrations showing which window was called what in the manual. I'm sure the reason for the lack of pictures is that the manual's for all computer versions. Obviously a separate manual for each type of machine would have helped. Or at least a supplemental manual. But all you get on that order is a card that tells how to start the game and how to copy it. If I seem to be dwelling on the short coming of the manual it's because I feel it is the major weakness of the package. I even fear some people may never get into the game as much as it deserves simply because it is so hard to dig out of the manual what is going on. For an example: in the tutorial it talks about building PDBs. Fine. What's a PDB? There is no glossary, no index, and PDBs have never been mentioned before. On the computer screen you can see the number of PDBs that you have. Farther on into the manual, Defense Bases are mentioned. Could PDB stand for Planetary Defense Base? The user is never told. After playing for awhile, you can figure out this must be the case, but for a beginner, it is only confusion. One last flame job on the manual and I'll move on to the play of the game. In a program of this complexity you need to have clear concise and orderly access to information. You should be able to look in an index or at least an area and find something about that subject. Half the game is explained in an unorganized narration during the tutorial (then never mentioned again), there is no reference section, and the other half of the game is explained under the advanced rules section. You have to read the whole manual then memorize where to find the info. Don't get me wrong, it's not *that* hard to get going in RFTS. I just feel you could get a much better start with better instructions.

Coneheads, Necromaniacs, Kzinti, And The Killer Klowns

You get four players in the game. No more, no less. The slots you can't fill with someone who has a pulse will be controlled by the computer. The computer can play as a beginner, experienced, or veteran. This may be in order of toughness or not. Again the manual doesn't say whether vets or experienced are better players. It doesn't matter in my case because, so far, the beginners are hard enough. Each player starts with one home world, no ships and $200 in the bank. (RFTS calls money, "Resource Points" or RPs. I'll use the "$" sign here because it's how we think.) There are two kinds of turns in the game. The first is called the 'production' turn, and the second is the 'movement' turn. The first turn is a production turn and of course it's year number one. Every other year you will get to set how you spend your money for that turn. The turns in between (movement only turns) you can only move ships around and colonize, attack or whatever, but not spend money. You start with your home worlds production window open and the planets' status window open. You punch in the amounts you wish to spend on your people and PDBs, as well as how many ships you want to buy and how much to invest in technological research. If you don't have a clue as to how to spend your money, you can ask the computer for advice. After you see what the computer recommends, you can change the values to whatever you like. As the game progresses and you colonize more and more worlds, you will have to cycle through them one by one to spend your money. You can invest in technology (to build better ships), starve the populous (inducing riots and rebellion, thereby shrinking the population back down to a more easily subjugated mass), build transports (to conquer and populate other worlds), save the planet's environment from industrial pollution, and so on. You really can get involved with the health of your planets, and it is pretty crushing to watch your pet world blow into a supernova after you've saved it from plagues, riots, and alien invaders.

After you spend your money (you can save it for later if you want) you end the production phase for the year. All the ships and spending on the planet for defense bases and what not take place immediately. Then you go into the movement phase. This is when you enter in where you want your task forces to go, where to colonize, etc. The whole procedure can be done through point and click. This includes the production phase as well (except for entering the $ amounts). The player who spends his money the wisest will usually end up winning. But, just like life, sometimes you don't get a fair shake. Every now and then one of your stars may go nova, you may get plagues, earthquakes, and of course the computer always wipes out your task force a turn before you can jump on the alien who's trying to kill off the other player. After each player has spent his money and made his movements, the computer executes all the moves simultaneously.

The players can nuke planets, colonize planets, invade other players planets', and generally wreak havoc amongst one another. This cycle goes on until only one player is left, or one player has an overwhelming amount of victory points.

The Outerspace Interface

This is probably a good point to rave about the user interface of RFTS. It is outstanding! SSG has used the desktop interface to its fullest! You want to know about a star and its planets? Click on the beautiful, *animated* desktop map (which is drawn in 640 mode), and poof! A window pops up and tells you whether or not you've explored it, how many ships you have there, how many defense bases etc. Everyone I've played this game with has been really impressed with the interface. It is *very* well done. To top it all off, RFTS fully supports all of your New and Classic Desk Accessories. (In fact, I wrote this review with *GS+* Magazine's EGOed New Desk Accessory while playing RFTS.) I have nothing but praise for the way this game looks and plays. I can almost imagine the authors seeing the IIGS for the first time and saying "This is the machine we've waited for all these years. Now we can finally write Reach for the Stars." Seriously, the interface shines!

The Fly In The Soup

Of course there is a problem or two. This game can take many many sessions to complete, it is almost completely open ended. The object is to overwhelm the other players, but you can keep on playing and give them a chance to recover, with a game possibly running for as long as hundreds of years (turns). This isn't a problem, but to play for so long you have to save the game. You can save as many games as you have disk space for and restart any of them whenever you want. (I love save game features like this.) But, there's a little problem here. When running from a hard disk like I do, you must have your system setup *exactly* the same as it was when you first started the game. (Say what?) Let me give an example: Say you get a New Desk Accessory and put it on your system disk. Then you play RFTS and save the game. Later you go back to play again, and provided you have the same amount of memory used up by your NDAs, CDAs, Inits and so on, the game will work perfectly. But, if you decide later you don't want that new desk accessory any more, and throw it out, the next time you run RFTS your setup will be just a little different and RFTS won't like to run your saved game so well. When you come up in the game your players names will be missing or trashed and a couple of the minor windows will also have some junk in them. I suspect this has something to do with the fact that it came with a slightly older version of GS/OS than my hard disk has on it. Once you start playing though, the names appear correctly everywhere else, and it seems to have no real affect on game play. If you find this happening and you can remember what you had in your system before, you can go back and try restarting again after you've put that desk accessory back and the saved game will run like a champ. Of course if you always boot from the game disk or make a special boot disk for the game, you'll never have this problem. My machine only has 1.25 MB of RAM in it. This may also be the problem.

The Final Word

Reach for the Stars doesn't have graphics and music like Rastan or The Immortal, but it's not that kind of a game. Is this game worth buying? *Yes*. For a game of this quality, and outstanding playability, I would have gladly paid twice the cost. Of course you may not like "Conquer the Galaxy" type games as much as I. That's up to you. But, if you do, then get Reach for the Stars. SSG has moved to the front of the pack in "serious" game design in my book. GS+

THE FINE PRINT IN OTHER ADS:

"Not responsible for product compatibility"
"Add 3% (minimum $4.00) for each shipment"
"Personal and company checks allow 3 weeks to clear"
"20% restocking charge on returned items"
"All sales final."

THE FINE PRINT IN OUR ADS:

"We Pamper Apple Computers"
"Trade in credit for old hardware"
"85% of our business is the Apple II line"
"Hardware - Software - Repairs - Training"
"We use what we sell"
"Free shipping"

Some of the company we keep:

American Micro Research 🍎 Applied Engineering 🍎 Appoint 🍎 Apricorn 🍎 Beagle Bros 🍎 Broderbund 🍎 Brother International 🍎 CH Products 🍎 Data Spec 🍎 Electronic Arts 🍎 Learning Company 🍎 Logitech 🍎 Olympia 🍎 Orange Micro 🍎 Qtronix Corporation 🍎 Roger Wagner 🍎 Seikosha 🍎 SoftSpoken 🍎 Supra Corp 🍎 TimeWorks 🍎 Vitesse 🍎 Westcode Software 🍎 Wordperfect Corp 🍎 Zip Technology 🍎 Zoom 🍎

Some of the items we carry:

cables 🍏 chips 🍏 computers 🍏 cooling fans 🍏 data switches 🍏 disks 🍏 dot matrix printers 🍏 drive boards 🍏 fax & modem switches 🍏 5.25 drives 🍏 hard drives 🍏 interface cards 🍏 joysticks 🍏 keyboards 🍏 laser printers 🍏 mice 🍏 modems 🍏 monitors 🍏 mouse pads 🍏 network connectors 🍏 power supplies 🍏 printer cards🍏 ram boards 🍏 scsi cards 🍏 software 🍏 surge protectors 🍏 3.5 drives 🍏 track balls 🍏

LEARNING EXPERIENCES
New Milford, CT 06776-1537
Phone: (203) 354-3669
Fax: (203) 355-1900
America Online: PAX5

HyperActivities

By Steven W. Disbrow

A Major Problem

You know it, I know it, we all know it: HyperCard IIGS is slow. It's not that surprising though, HyperCard IIGS is an amazingly complex and powerful program. But, all that power doesn't mean anything during the 5 or 10 minutes (or longer!) that you might have to wait to sort a stack.

As I've mentioned before, we keep our subscriber database in a HyperCard IIGS stack. (From here on out, I'm going to refer to HyperCard IIGS as "HyperCard." We all know which machine we are talking about, don't we?) When I first began to investigate moving this information from DB Master Professional to HyperCard, the first thing I noticed was that the sorting facilities of HyperCard became amazingly slow after you added about 50 cards to a stack. Our customer database has several *thousand* records in it, so I knew that I had to do something to speed things up or I'd be stuck in DB Master for the rest of my life. (DB Master is a great database, it's just pain in the rear to use.)

Out in the "real-world," this sort of problem doesn't come up very often. Most of the "big-name" database systems have options to insert new records into a database so that they are in the correct place—no sorting is necessary. DB Master does this sort of thing with ease (no pun intended). In HyperCard, though, when you say "New Card," that new card is inserted immediately after the current card—there is no built-in provision for inserting new cards into a stack based on some predefined-defined sort criteria. This means that you have to go back and sort the stack after you have finished adding all of your new cards.

Simple Solution

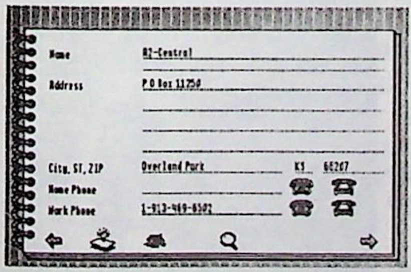
Clearly, this is not something that can be tolerated in a database with several thousand records. Fortunately, HyperCard has a programming language, HyperTalk, built into it that can be used to overcome this problem. The solution is to write the appropriate HyperTalk scripts that will automatically insert each new card into the stack in its sorted position.

To illustrate one possible way to do this, I have created a self-sorting address book stack and placed it on your *GS+* Disk. (The file name is **SSAddress.Book.** See "Using The *GS+* Disk" for more information.) "Self sorting" means that when you create a new address book entry and type in the name for that entry, the entry is automatically moved to its sorted position in the stack. But, before we get into the technical mumbo-jumbo, let's see how we go about . . .

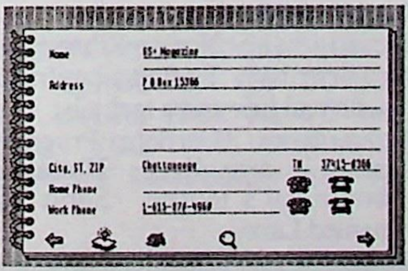
Using The Address Book

As provided on your *GS+* Disk, the address book already contains six entries containing the names, addresses and phone numbers of several Apple II publications. These are shown below. (Sorry about the size . . .)

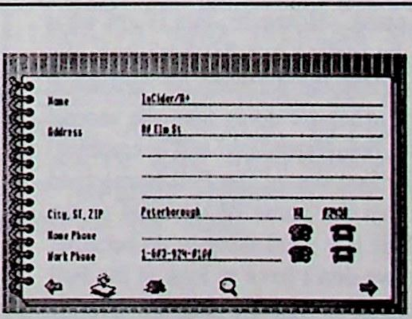
To enter a new record, simply press Command-N (for New Card). An empty address book card will be created and the cursor will be on the first line of the name field. As an example, let's say you wanted to enter a record for "Ginger Grant." Type "Ginger" on the first line of the name field and press the return key. Now type "Grant" on the second line of the name field and press return. As soon as you press return on the second line of the name field, the address book begins to look for the correct position for the card. When it finds that position, it moves the card there. When it is finished, the cursor is placed on the first address line so you can finish typing in the address information in the remaining fields. Those other fields include four separate address fields (to provide space for those friends you might have in foreign lands),



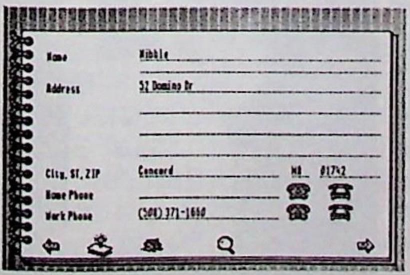
Name A2-Central
Address P O Box 11250
City, ST, ZIP Overland Park KS 66207
Home Phone
Work Phone 1-913-469-6502



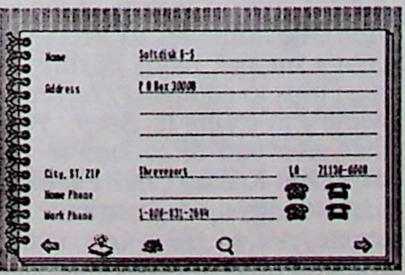
Name GS+ Magazine
Address P O Box 15366
City, ST, ZIP Chattanooga TN 37415-0366
Home Phone
Work Phone 1-615-870-4960



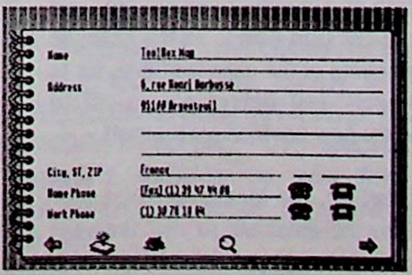
Name inCider/A+
Address 80 Elm St
City, ST, ZIP Peterborough NH 03458
Home Phone
Work Phone 1-603-924-0100



Name Nibble
Address 52 Domino Dr
City, ST, ZIP Concord MA 01742
Home Phone
Work Phone (508) 371-1660



Name Softdisk G-S
Address P O Box 30008
City, ST, ZIP Shreveport LA 71130-0008
Home Phone
Work Phone 1-800-831-2694



Name ToolBox Mag
Address 6, rue Henri Barbusse
95100 Argenteuil
City, ST, ZIP France
Home Phone (Fax) (1) 39 47 94 08
Work Phone (1) 30 76 18 04

City, State, Zip and a home and work phone number. Pressing return at the end of each line will advance the cursor to the next line. If you ever have to change the name of someone in your address book, just go to the appropriate card, click on the name field and start typing. As soon as you hit return on the second line, the card will be moved to its new position.

There are also several icons on each card. Most of these icons do exactly what you would expect them to do (move from one card to the next, go home, etc.) However, next to the Home and Work Phone fields there are two icons, one of a yellow telephone, and the other of a black telephone sitting on top of a modem. If you click on one of the yellow telephones, HyperCard will dial the appropriate number through the IIGS speaker. If you click on one of the black telephones, HyperCard will dial the number through your modem (if you have one attached.) Now, on to the technical . . .

Mumbo-Jumbo

Before you can sort a database (or a HyperCard stack—I will use the terms "database" and "stack" interchangeably because they really are the same things), you have to decide what you will be sorting on. For an address book, you want to sort the entries by name. In this case, the name is stored in a background field called "Full Name." The first line of the field contains the first name, and the second line contains the last name. However, this is not what we will be sorting our database by. We will be sorting the database by *card name*. Basically, what we do is take the second line of the Full Name field and concatenate it with the first line of the Full Name field and then set the name of the card to the result. This is a good thing to do because, this way, we don't have to look at the Full Name field when we are trying to determine the position of a newly inserted card—we just look at the card name. In the example above, the name of the card would be set to "GrantGinger" just before we go off to determine the new position for the card.

As I said before, the idea is to insert new records into the database so that they are already in their sorted positions. This stack uses a slight variation on that idea: new cards are placed in the stack in standard HyperCard fashion (i.e. wherever you happen to be when you hit Command-N), and then moved to their sorted position when sufficient information is available to find the cards sorted position (i.e. when you finish typing in the name for the card). The problem is: how do we tell when we have enough information? The answer is simple, we trap the return key!

In its simplest form, the HyperTalk code to trap the return key (and do something useful with it) looks like this:

```
on returnInField
  tabKey
end returnInField
```

This small handler is called every time the return key is pressed in a field. In response, the script sends out the message `tabKey` which causes HyperCard to move the cursor to the next field. This makes HyperCard much easier to use for data entry.

As I said however, this is the simplest case. What *we* need to know is when the return key is pressed *on line 2 of the Full Name field*. When this happens, we should have enough information (a first and last name) to find the proper position of our new card in the stack. The handler for this is quite a bit larger than the one above, so, instead of presenting it here, I'll just tell you to look at the script of the background for the self sorting address book. (Press Command-Option-B to view this script after you open the stack.)

After we have determined that we have enough information, and set the name of our card to the name of the person we are entering into the address book, we call the `findPosition` function to determine the sorted position for the card. Since the contents of the stack are already sorted, `findPosition` uses a simple Binary Search method to determine the sorted position for the card. In addition to being simple in concept, a Binary Search is also very fast. In fact, it takes about the same amount of time to insert a record into our customer database (which has several thousand records) as it does to insert a record into the address book stack. When `findPosition` is finished, it returns a number telling the correct position for the card to the handler that called it. The card can then be moved to the new position.

That's All There Is To It

If it seems like I'm being just a bit vague, I am. The fun of HyperCard is getting into stacks and pulling them apart to see how they work. So get in there and start clicking on things!

In our next HyperActivities installment, we'll take a look at how to generate custom reports, without using templates! **GS+**

KANSASFEST BABE DATA SHEET

Name: Noreen Ribaric

Ambition: To find a boyfriend with real hair and a real job.

Turn-Ons: Garfield. Programming in a negligee. Hiding my boyfriend's toupee. Short guys named Lane.

Turn-Offs: People that use the lumbar support feature of the Ford AeroStar minivan to accompany songs on the radio during long road trips to computer conventions while I'm trying to sleep. People that give bad directions to pizza parlors in Kansas City. People that make things up and attribute them to me just to fill a few stinking columns in their stupid magazine.

Warranty Disclaimer

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

IN NO EVENT WILL APPLE, ITS DIRECTORS, OFFICERS, EMPLOYEES OR AGENTS 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 APPLE SOFTWARE EVEN IF APPLE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU. Apple'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.

New Products

Compiled By Steven W. Disbrow

The following product descriptions were taken from press releases that we have received here at *GS+* Magazine and that we were given at KansasFest. (Actually, there were lots more new products announced at KansasFest, these are just the ones that we got press releases for.) As with all press releases, the products described here may not actually exist yet.

Liberty!

Micol Systems of Canada, makers of Micol Advanced BASIC, announced a new 3.5-inch disk controller/drive combination that will allow IIGS, IIe and II+ owners to store up to 1.6 MB on a high-density floppy. Since Liberty includes a controller card, you can actually use it to boot from a 1.6 MB floppy! (Unlike the Applied Engineering 3.5-inch high density drive.) Liberty also includes software that will allow you to directly read and write MS-DOS diskettes! With the apparent omission of an MS-DOS FST from System Software v6.0, this could be just the thing for folks that need greater connectivity to the MS–DOS world.

The price for the Liberty disk controller and a single 3.5-inch drive is a measly $249. For more information, contact:

Micol Systems
9 Lynch Road
Willowdale
Ontario, Canada M2J 2V6
(416) 495-6864

Internal Changes

After Ingenuity bit the dust, the prices on Vulcan internal hard drives began to slowly creep upwards. Fortunately, another company has entered the fray with an internal hard drive. ECON Technologies has introduced its "Pegasus" line of internal, *SCSI* hard drives for the IIGS. Yep, you heard right, these drives can use either the RamFAST or Apple DMA SCSI cards to communicate with your computer. A special SCSI cable allows you to add more SCSI devices to you computer without taking up an additional slot.

The Pegasus drives are available in 200 MB, 100 MB, 50 MB and 0 MB configurations. Yep. You read that right too. If you want, you can buy the Pegasus drive in kit form and add your own 3.5-inch hard drive mechanism!

As if *that* weren't enough, the Pegasus drives come with a complete set of disk utilities. Not just a disk formatter, but a file recovery utility, a backup utility and an optimizer!

If you still aren't convinced that these guys are serious, they have a 30-day free replacement policy and a two-year limited warranty. The prices aren't that bad either: 0 MB - $299. 50 MB - $599. 100 MB - $769. 200 MB - $1099. For more information, contact ECON at:

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

Dream On

You've heard about it for months, and apparently, it's about ready to go. DreamGrafix is a 3200 color paint program that can produce some incredible pictures. DreamGrafix supports all current IIGS graphics formats and can work in any screen mode. Best of all, it's not copy protected, installs on a hard drive and is completely compatible with NDAs, CDAs and everything else that makes the IIGS the IIGS. The price is $99.95. For more information, contact:

DreamWorld Software
P. O. Box 830
Iowa City, IA 52244-0830
(319) 338-6491

Your Own "Private Investigator"

DataComb has just introduced a New Desk Accessory called *File Detective*. File Detective allows you to search for files that are hidden away in that mess of folders on your disks. File Detective allows you to search for files based on name, type, and whether or not a file has a resource fork.

File Detective can also look *inside* files to search for strings. File Detective can even search the insides of resource forks.

File Detective requires System Software v5.0 or later (v5.0.4 is recommended). It costs $20 and includes a 20-page user manual, and has a 30-day, money-back guarantee. For more information, see the "*GS+* Classifieds" in this issue or contact:

DataComb
1310 Cholla Court
Lake Havasu City, AZ 86403

Believe It Or Not . . .

The Byte Works has introduced ORCA/*Integer BASIC*. Yes, you read that one right too. Now you can take all those old Integer Basic programs you have lying around and, compile them to run under GS/OS.

ORCA/Integer BASIC generates 65816 code and you can use the PRIZM source level debugger to step through your programs. You can even link in routines written in other ORCA compatible languages.

Just imagine, you could rewrite "Hunt the Wumpus" to use the desktop! Speaking of "Hunt the Wumpus," if you order ORCA/Integer BASIC before September 30th, you get a 3.5-inch disk packed with classic Integer BASIC programs! Programs like Eliza, MoonLander, TimeBomb, and MasterWoz!

Finally, ORCA/Integer BASIC includes *complete source code* to the ORCA/Integer BASIC compiler! If you've ever wanted to see how a compiler actually works, this is the product for you!

For more information on ORCA/Integer BASIC contact the Byte Works at:

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

GS+

GS+ Classifieds

Advanced Printing Concepts

300 DPI Laser Printing With **Enhanced Resolution.** Send us your Apple II data via disk, modem, or fax. Overnight and authorial services available! $1.00 per page for basic service. Discounts for large quantities. For more information **call (708) 215-8466** 9 am to 5 pm central time seven days a week.

Send US mail to:
Advanced Printing Concepts
PO Box 97
Wheeling, IL 60090-9998

Please include $2.50 for shipping and handling. Unique fonts must be included. Prices subject to change.

New!

The File Detective 1.0 (TFD). It can make your life easier and save you time because TFD can: (1) Search for up to 64 clues with wildcard matching. (2) Target for search or catalog a file, directory, directory tree, disk, or all disks on-line. (3) Recognize resource forks. (4) Limit file-types to access. User can customize feedback. Requires IIGS with System Software v5.0 or later. Find out what you've been missing! A very useful desktop program for only **$20.00**!

Contact: DataComb
1310 Cholla Court
Lake Havasu City, AZ 86403

3.5-Inch Disks

Great Quality. Lifetime Guarantee. Only **$0.25** each. Add $1.00 for shipping for every 10 disks. Send check or money order to:

David Coher
4213 Hackett Ave
Lakewood, CA 90713

1/2 Price Software And Hardware

Questmaster - $20
Music Construction Set - $10

Transylvania III - $20
GS RAM Plus (0 MB) - $75

Art & Film Director - $10

Contact: *GS+* Magazine
c/o EGO Systems
P.O. Box 15366
Chattanooga, TN 37415-0366
(615) 870-4960

Readers can place an ad in the *GS+* Classifieds for only $5. This cost buys 25 words in one issue of *GS+*. Additional words are just 10 cents each. The *GS+* Classifieds are the perfect way to contact all of the other IIGS owners out there. The deadline for inclusion of a classified ad in the next issue (Volume 3, Number 1) of *GS+* is September 15, 1991. Simply fill out a photocopy of the coupon below; or send your ad along with your name, address, phone number, number of issues to run, and payment (made payable to EGO Systems) to us here at *GS+*; or call us at (615) 870–4960, Monday through Friday between 9 a.m. and 6 p.m. EST, to place an ad with your MasterCard or VISA.

GS+ Classified Ad Order Form

Ad copy: ____
____
____
____
____
____

Number of issues to run: ______ Number of words: ______ Total enclosed:$ ________
Name: ______________________________ Phone: ________________
Address: ____
City: ______________________________ State: ______ Zip: ____________

Product Updates

compiled by the **GS+** Staff

There certainly were a lot of product enhancements announced at KansasFest. But, space is running short, so we thought we would hit the biggies—in order of increasing version numbers.

HyperCard IIGS v1.1

Apple is continually updating Macintosh HyperCard (they usually have to do a new version for each new Macintosh), and they seem to have the same thing in mind for HyperCard IIGS. HyperCard IIGS v1.1 will be sort of a "we've had the time to clean things up internally, so here it is" type version. In other words, there won't be much new stuff, but what's already there will be cleaned up to be more robust and yes, slightly faster. For example, when you click on a button in HyperCard IIGS v1.0, HyperCard writes the button's new state to disk. This takes time, so in version 1.1, it's been eliminated.

There *will* be a few new things though. The HyperCard IIGS team has decided to take some really nifty things from HyperCard v2.0 for the Mac and put them into HyperCard IIGS v1.1. They have also decided to add a few new things that are specific to the IIGS. Like support for all of the new features of System Software v6.0.

The price of HyperCard IIGS v1.1 should remain the same as v1.0 and it is expected to ship when System Software v6.0 ships later this fall. No update or other information was available.

ORCA/M v2.0

Mike Westerfield is a busy man. Lately, the main reason he's been busy has is all the work he's been doing on ORCA/M v2.0. This latest version of the venerable Macro Assembler and development environment looks to be a real winner.

First, let's run down some of the features of the new Shell:

Full support for GS/OS pathnames and prefixes, all GS/OS FSTs and AppleTalk NetWorks.

Support for Object Module Format (OMF) v2.0 files as well as older OMF v1.0 files.

Longer shell variable names and command lines (up to 64K)!

The ORCA editor has been rewritten and looks to be a drastically improved program. Some of the new features are:

Files can be any length up to the size of available memory.

The editor now supports long lines (greater than 255 characters) and tabs!

Up to 10 files can be open at once! Hurray!

Mouse support has been added, although you can still use the keyboard exclusively if you wish.

The search and replace functions have been changed to search for tabs, spaces and even do a case sensitive search.

You can now shift blocks of text to the left or right. Great for "structured" programming (at least on screen).

And finally, ORCA/M v2.0 will contain two (or three, depending on how you count) of the most useful IIGS programming utilities: GSBug and Rez/DeRez!

The retail price for ORCA/M v2.0 will be $125. Current ORCA owners can upgrade for $40. Owners of any other Apple II assembler can upgrade for only $62.50. No mention as to when ORCA/M v2.0 will be available, so, to find out, contact:

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

HyperStudio v3.0

Opening up the A2-Central Summer Conference at KansasFest on July 18 was Roger Wagner and his presentation entitled "New Horizons for the Apple IIGS." After a cute video of how hypermedia has evolved from AppleVision to HyperStudio, Roger launched into his presentation of HyperStudio v3.0, which has a tentative release date of September 15.

One of the main improvements in HyperStudio v3.0 is the more efficient use of memory. To accomplish this, HyperStudio v3.0 will make extensive use of resources and compressed sounds, shared icons and shared sounds.

Another improvement in HyperStudio v3.0 is that is will now be much easier to use. In the graphics category, it now supports 320-mode graphics, color cycling, 256-color graphics, and tear-off menus! Buttons are now editable, can have new shapes, and can be icons! Improvements in stack navigation include using marked and named cards and "jumping" to any card in any stack.

According to Roger, HyperStudio v3.0 has a "bzillion" new features including "more button tricks" (Roger's new, preferred, term for XCMDs), nearly 100 callbacks, and . . . a scripting language! Most excellent!

There are actually about 5 pages worth of new features in HyperStudio v3.0. Of course older HyperStudio stacks will be compatible with HyperStudio v3.0. New purchasers should expect to pay $169 (increased from $149) for HyperStudio v3.0. The upgrade price for version 3.0 will be around $40 or $45—but this is for an entirely new package with new manuals. If you purchase the current version of HyperStudio within 60 days of the release of version 3.0, you will only have to pay the difference of $20 to upgrade to version 3.0. For more information, contact:

Roger Wagner Publishing, Inc.
1050 Pioneer Way, Suite "P"
El Cajon, CA 92020
(619) 442-0522

GS+

Contest #4 Update

By Steven W. Disbrow

If you have not heard by now, and if the little ™ symbol on the cover of the magazine didn't give it away, we are going to continue using the name *GS+* Magazine! How did this happen? Good question.

If you will remember, in our last installment, I mentioned that I had written a letter to John Sculley, CEO of Apple Computer, Inc. I also mentioned that I had not had a reply to my letter as we went to press.

In my letter I explained our situation to Mr. Sculley and asked him if there was something he could possibly do for us. Apparently, there was. Mr. Sculley passed the letter along to Jill Sarnoff, Apple's Senior Trademark Council, who took the necessary measures and then called me to inform me that we could continue to use the name "*GS+* Magazine" without any further hassles.

The upshot of all this is that, surprise, *Apple really does give a damn about the IIGS market!* Combining this surprising revelation with all of the really neat stuff we saw at KansasFest gives me the feeling that the IIGS market isn't going to be such a bad place to be in the next few months. Hey! I'm actually excited!

So What About The Contest?

Well, after we found out we would be able to continue using *GS+* I decided that we still had to give away some sort of prize. So, we put all of the entries into a big box, and Susan Thoeming drew out three winners. The prizes were a half-year, full-year and two-year subscription extensions.

The Winners

Third place, and a three-issue extension of his *GS+* subscription, goes to Elliot Chubb of Sunset Hills, MO.

Second place, and a full-year extension of his *GS+* subscription, goes to Martin Sheehan of Plymouth, MN.

First place, and a two-year extension of his *GS+* subscription, goes to Kenneth P. Hass of Newbury Park, CA.

Thanks!

This brings Contest #4 to an end. (Finally!) My sincere thanks to everyone who entered this contest. Even though we ended up right back where we started, it was a real treat reading all of the suggestions we received. If you have an idea for a new contest, be sure to let us know about it.

An extra special thanks, to John Sculley and Jill Sarnoff at Apple Computer, Inc. If you were going to write Mr. Sculley a letter asking him to let us continue using the name *GS+*, how about writing him a thank-you letter now that he has! **GS+**

GS+ User Group Connection

By Steven W. Disbrow

Merger Mania

In the last few months, we have gotten several newsletters from Apple II groups that are having to consider the ugly realities of dwindling memberships and member apathy. Many of these groups are considering (hold your breath) merging with their local Macintosh user groups in order to survive. Of course, some members of these groups are vehemetly opposed to such a manuever while others see it as inevitable.

At this point, I don't know what advice to give on this subject. Our local group, while not the most lively or energetic bunch, is not in the same dire straights as some of these other groups. However, I can empathize with the leaders of these groups that have to face the fact that it is *very* difficult to get the membership involved, let alone bring in new members.

So I'm asking for your opinion. What do you think these groups should do? Join with the Macintosh users? Continue to go it alone? Call it quits? Send your ideas to us here at the *GS+* User Group Connection and we'll share them with all the user groups out there.

That's all I have room for this time. Until later, remember, your user group needs *your* help! Volunteer today! **GS+**

We Want You!

If your IIGS user group or IIGS special interest group (SIG) is not a member of the *GS+* User Group Connection, we want to change that. To become a member, simply have one of your club officers contact us. All we need is the name and address of your group. However, if you give us a free subscription to your group's newsletter, we'll give your group a free magazine-only subscription to *GS+* and access to the *GS+* User Group Connection reprints! Send that information and/or newsletter subscription to:

GS+ User Group Connection
P. O. Box 15366
Chattanooga, TN 37415-0366

Feedback

How did you first hear about *GS+* Magazine?

Are you a subscriber to *GS+* or was this a sample issue?

If you are a subscriber, do you receive the Magazine and Disk? If you do not receive the disk, why not?

How would you describe your level of computer experience?

- ❑ babe in the woods
- ❑ novice
- ❑ I get by
- ❑ fairly proficient
- ❑ experienced
- ❑ power-user
- ❑ digital deity

Are you a member of a user group?

- ❑ Yes
- ❑ No

Please tell us a little about your IIGS system. Do you have a:

- ❑ Hard drive
- ❑ Modem
- ❑ Dot Matrix Printer
- ❑ Laser Printer
- ❑ 3.5-inch drive
- ❑ 5.25-inch drive
- ❑ Scanner
- ❑ Digitizer
- ❑ Accelerator card
- ❑ Fan
- ❑ Extended Keyboard
- ❑ Anything else?

How much memory do you have in your IIGS?

Is your IIGS part of an AppleTalk network?

- ❑ Yes
- ❑ No

Do you have a SCSI card in your IIGS?

- ❑ Yes Which one?
- ❑ No

On a scale of 1 to 5 (with 1 being "poor" and 5 being "excellent"), how would you rate the following items from this issue of *GS+*? (If you don't receive the disk, you don't have to rate the disk based items.)

Roger Wagner Interview _____

Working With The Toolbox _____

Watchdog _____

MacZombies _____

KansasFest Report _____

A Peek At System 6 _____

Programmer's Queue & A _____

HyperActivities _____

How would you rate the reviews in *GS+* Magazine? _____

How would you rate *GS+ Magazine*? _____

How would you rate the *GS+ Disk*? _____

How would you rate *GS+* overall? _____

How often do you think we should update our programs?

- ❑ Every issue
- ❑ Every other issue
- ❑ Only when the update is a significant improvement
- ❑ Never

Should we review more:

- ❑ Educational Software
- ❑ Games
- ❑ Hardware
- ❑ Productivity Software
- ❑ Utilities

How would you rate the technical content of *GS+* Magazine?

- ❑ Child's play
- ❑ Just right
- ❑ I'm drowning in jargon!

What would you like to see in the next issue or two of *GS+*?

We want to have the best, most reliable advertisers in the business. Who would you recommend that we try to get? Why?

If there was one thing you could change about *GS+*, what would it be?

Do you think you will renew your *GS+* subscription? If not, please tell us why.

Anything else you want to say? Feel free to add additional sheets.

Send this sheet (or a photocopy) to:
GS+ Feedback
P. O. Box 15366
Chattanooga TN, 37415-0366

GS+

Buying Ad Space In GS+ Magazine

1/3 page - $30
1/2 page - $45
2/3 page - $60
1 page - $75
2 pages - $140

If you want your ad to appear on the back cover or inside back cover, add an additional $75.
Save an additional 20% by placing the same size ad in more than one issue!
Prices good through December 15,1991.
Request your ad space today to guarantee these low prices!

DEADLINES:
September-October 1991 (Volume 3, Number 1) August 31, 1991
November-December 1991 (Volume 3, Number 2) October 15, 1991

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
c/o **EGO** Systems
P.O. Box 15366
Chattanooga, TN 37415-0366
Voice phone: (615) 870-4960

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.

GS+ Ordering Information

GS+ is published bimonthly and sold for $3.00 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, c/o EGO Systems, P.O. Box 15366, Chattanooga, TN 37415-0366.

Name: ____________________ Phone: (   ) ____________________
Address: ____
City: ____________________ State: __________ Zip: ______________

- ❑ 1-year subscription - magazine + disk - $36
- ❑ 1/2-year subscription - magazine + disk - $20
- ❑ 1-year subscription - magazine only - $15
- ❑ 1/2-year subscription - magazine only - $8
- ❑ Sample issue - magazine + disk - $8
- ❑ Sample issue - magazine only - $3

- ❑ Check or money order enclosed
- ❑ Bill my MasterCard
- ❑ Bill my VISA

Card No.: ________________________________
Expiration Date: ____/____
Signature: ________________________________

If you prefer to use your credit card to order by phone, give us a call at (615) 870-4960, Monday through Friday, between 9 a.m. and 6 p.m. EST. All subscriptions will start with the next issue published. Please allow 2-8 weeks for delivery of first issue. Tennessee residents add 7.25% sales tax. Add $1.50 an issue if you want First-Class delivery. Add $1.50 an issue for delivery to Canada or Mexico. Add $1.50 for surface mail to all other foreign countries. Add $5 per issue for Air Mail to all other countries.

TMS Peripherals

Authorized Quantum Dealer

Great service—this is the first company to deliver on time. I was amazed."

—Mike Wyatt

"I love the drive. Delivery and service are impressive. I had the equipment 26 hours after I ordered it. Thanks!"

—Mike Seaton

"...they absolutely excel in having the best technical service that I have ever encountered."

—Webster W. Plourd
Col. USAF (Ret.)

"If you need a hard drive, I highly recommend that you buy it from TMS. I was impressed with TMS' products and service...they truly deserve your business."

—Steven W. Disbrow
Publisher, *GS+* Magazine

WE ARE DEDICATED TO CUSTOMER SERVICE AND SUPPORT.

We're not just about great products at great prices, we're about excellent service too!

- Each *factory fresh* TMS drive is completely tested and verified by our technicians before it is shipped.
- TMS Peripherals' 30 Day Money Back Guarantee: *"If you experience a problem with your drive, TMS will ship you a new drive and pick up the suspect drive (at our expense), leaving you with virtually no down time!"*
- TMS Peripherals' Full 2 Year Warranty / Toll-Free Tech Support Line—if your problem persists after thorough troubleshooting by our expert technical staff—just ship us the suspect drive and we will send you a replacement unit within 48 hours.
- All Apple drives are preformatted and partitioned and come with the latest System Software & 10+ MB's of shareware. All Macintosh drives are preformatted and come with the latest System Software, 19+ MB's of BMUG shareware and DISK DOUBLER by Salient Software. DISK DOUBLER was rated 4 1/2 Mice as the 1990 MacUser Editor's Choice Award for Best Compression Product and doubles your Mac disk drive capacity easily and safely!
- Of course, each drive also comes with our enthusiasm for each of our products! For your convenience, we're open 8am–10pm (EST) 7 days a week and all products ship Federal Express. C.O.D.'s and P.O.'s accepted.

**All prices subject to change.*

FREE
FEDERAL EXPRESS SHIPPING
PLEASE CALL FOR DETAILS!

1.800.626.MEGS

1120 Holland Dr., Suite 16
Boca Raton, Florida 33487
Internat'l. 407.998.9928
FAX: 407.998.9983

VITESSE

"Setting a new standard in utilities for the Apple market." *InCider A+ '91*

SALVATION SUPREME

ALL 5 Salvation Utilities for the IIgs. *Only $89.95*

Salvation Utilities

This is the complete utilities series for IIgs users. Includes virus detection & cure, backup utility, hard disk optimizer, manager, and file repair & recovery.

Bakkup, Renaissance or Deliverance	$28.95
Exorciser	$27.95
Wings	$45.95

QUANTUM STORAGE SUBSYSTEMS

TMS PRO SERIES HARD DRIVES

	Int.	Ext.	
Pro 52 LPS	$249.95	$349.95	11ms
Pro 105 LPS	429.95	529.95	11ms
Pro 120	659.95	759.95	10ms
Pro 170	729.95	829.95	10ms
Pro 210	799.95	899.95	10ms

SYQUEST REMOVABLE HARD DISKS

TMS Pro R45, **$489.95** • 20ms • extra carts **$67.95**
TMS Pro R88, **$1319.95** • 20ms • extra carts **$139.95**

TAPE BACKUP SYSTEMS

The perfect backup solution for your TMS Pro Drive!
TMS Pro T60, **$439.95** • TMS Pro T155, **$499.95**

CHINON

TMS CD ROM, 350ms, only **$579.95**

PRO SERIES STANDARD FEATURES

- International Auto-Switching Power Supply
- Selectable Pushbutton SCSI ID Switch / VCA Mech.
- 2 SCSI Ports / 2 A/C Jacks / Shielded PS
- Thermostatically Controlled "Whisper Quiet" Fan
- 8+ MBs Shareware / Hardware & Manuals

Quickie Scanner by Vitesse

This is the only hand -held, grey-scale image scanner for all "slot equipped" Apple II's—the IIgs, IIe, II+ and the Laser 128. Digitize flat page images and save to disk in standard Apple II graphic file formats. ***Now only $195.95***

WESTCODE

Discover the power and ease of InWords today—InWords provides a revolutionary new way to enter text into your Apple II. Instead of slow tedious typing, InWords will input information at up to 500 words per minute. By simply passing your hand-held scanner over the page, you can create a word processor, data base, or spreadsheet file without typing a single word. Requires 512K and Quickie or LightningScan. **$79.95**

APPLE SOFTWARE

Apple Works 3.0	$165.95
Apple Works GS 1.1	$195.95

GLEN BREDON SOFTWARE

ProSel 16	$66.95
ProSel 8	$29.95
DOS.Master	$19.95

C.V. TECHNOLOGIES

Ram*FAST* SCSI v 2.0, **$189.95** or
$179.95 with the purchase of any drive!

GS-Memory Boards

0 Mb	$79.95
1Mb	$142.95
2Mb	$194.95
3Mb	$246.95
4Mb	$299.95

RAM Chips: 256k 100ns (set of 8), **$17.95**
1MB 100ns (set of 8), **$65.95**

ZIP TECHNOLOGY

ZIPGSX, 7 MHz/8k Cache memory, **$148.95**
ZIPGSX, 8 MHz/16k Cache memory, **$198.95**
ZIPGSX, 9 MHz/32k Cache memory, **$248.95**
***ALL ZIPGSX boards are DMA Compatible!**

Æ APPLIED ENGINEERING

Æ PC Transporter 768k	$249.95
Æ PC Transporter GS Kit	$36.95
Æ 3.5" 800k	$195.95
Æ 3.5" HD 1.6 Mb	$235.95
Æ DataLink 2400 Internal	$159.95
Æ DataLink 2400 Express	$159.95
Æ DataLink 2400 Express w/ MNP5	$189.95
Æ DataLink 2400 Express w/ MNP5/FAX	$215.95
Æ ReadyLink Communications Software	$56.95
Æ Conserver	$74.95
Æ HD Power Supply IIe	$74.95
Æ HD Power Supply GS	$84.95
Æ TransWarp GS 32k	$299.95
Æ TransWarp GS 32k Cache Upgrade	$79.95
Æ RamWorks III 0k	$119.95
Æ RamWorks III 256k	$129.95
Æ RamWorks III 512k	$149.95
Æ RamWorks III 1 Mb	$179.95

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