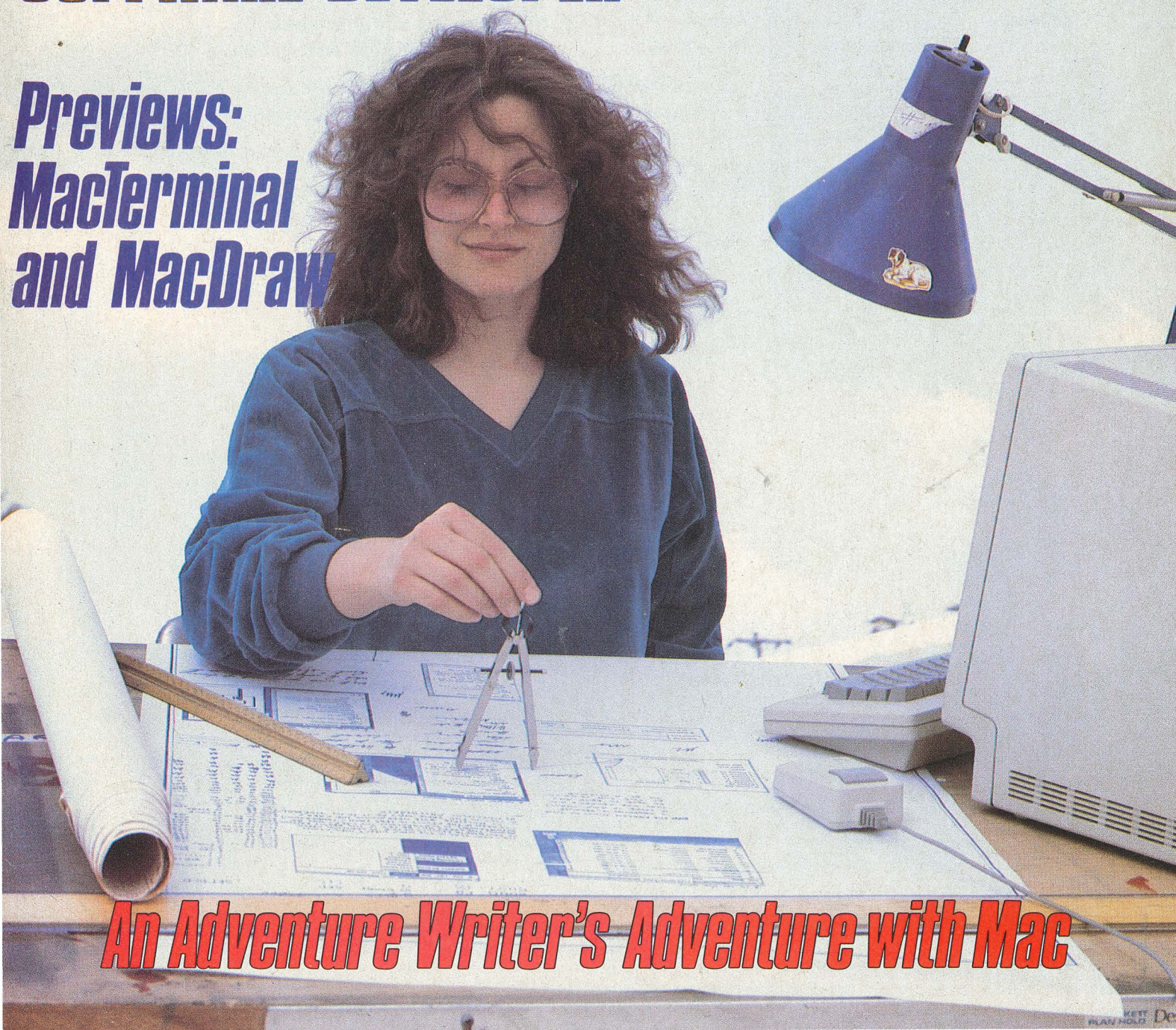
ST.MAC

Volume 1 June 1984 $2.00

HOW TO BECOME A MACINTOSH SOFTWARE DEVELOPER

Previews: MacTerminal and MacDraw



An Adventure Writer's Adventure with Mac

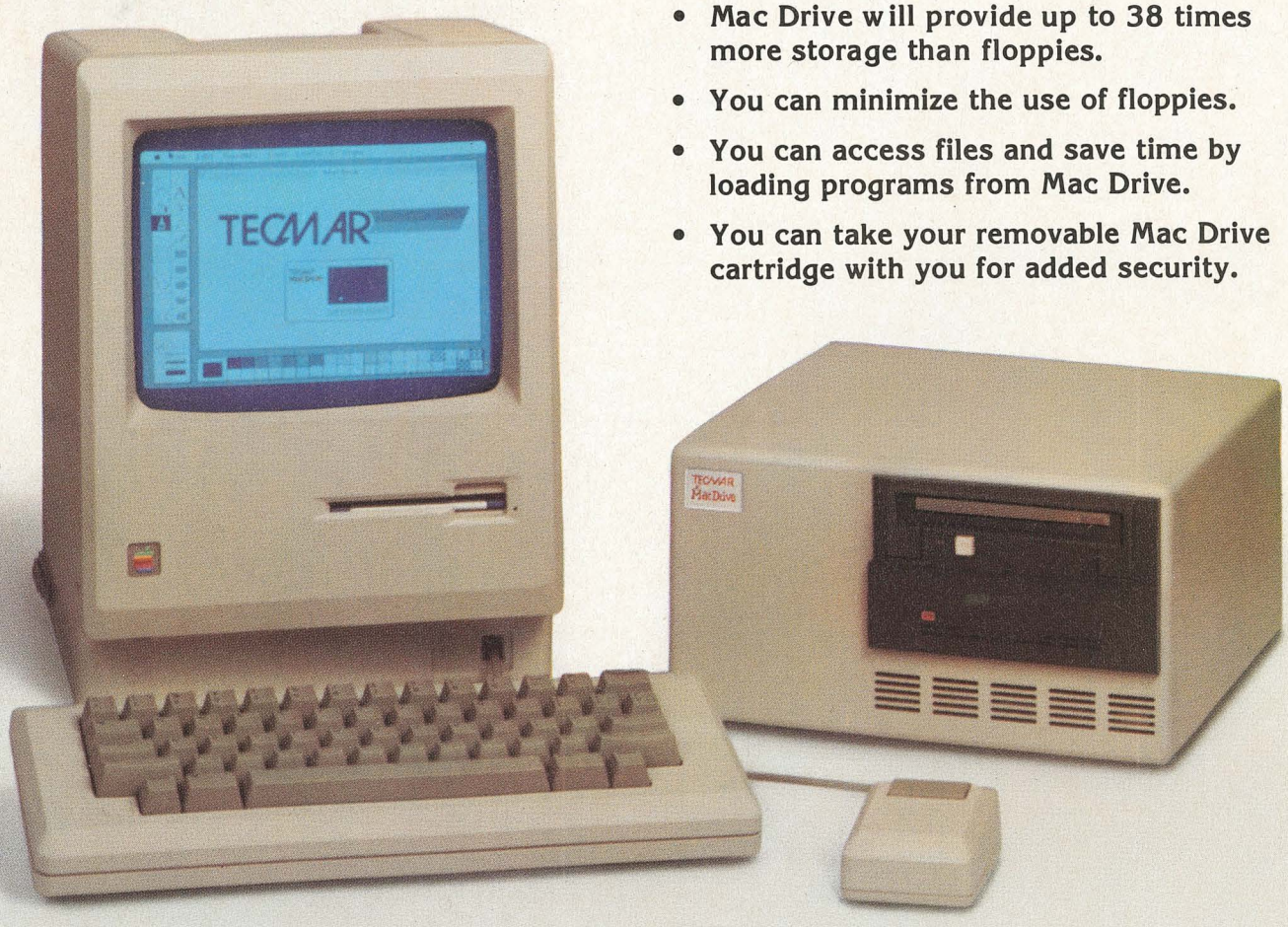
Tecmar Sweetens Macintosh® with Hard Disk Power

Mac Drive™

Tecmar's Mac Drive gives you a 10 megabyte fixed hard disk or a 5 megabyte removable hard disk. You can add a 5 megabyte removable hard disk to either of the above.

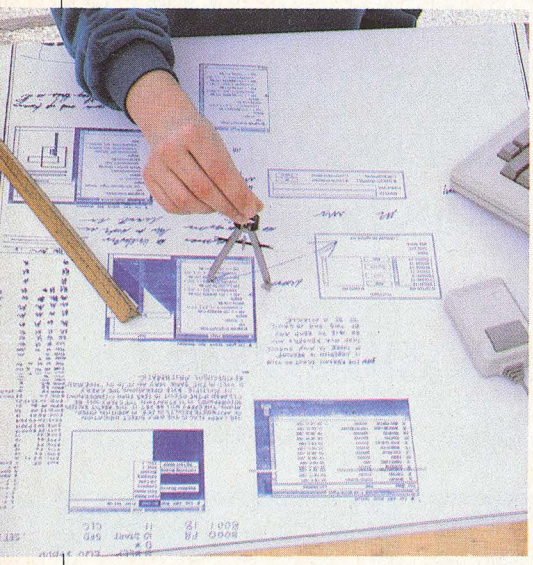
With Tecmar's Mac Drive . . .

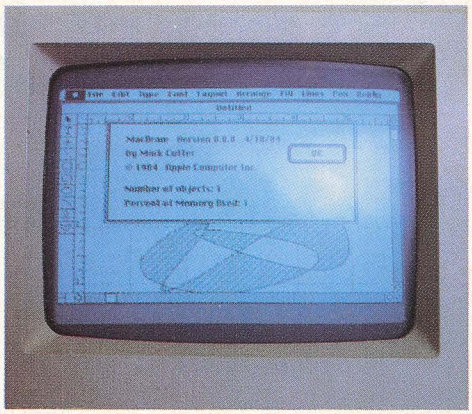
- Your Macintosh runs much faster than with the built-in floppy.
- Mac Drive will provide up to 38 times more storage than floppies.
- You can minimize the use of floppies.
- You can access files and save time by loading programs from Mac Drive.
- You can take your removable Mac Drive cartridge with you for added security.

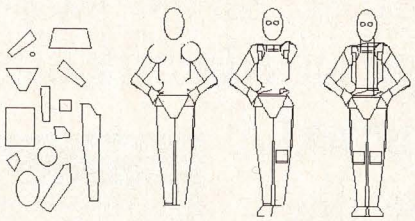


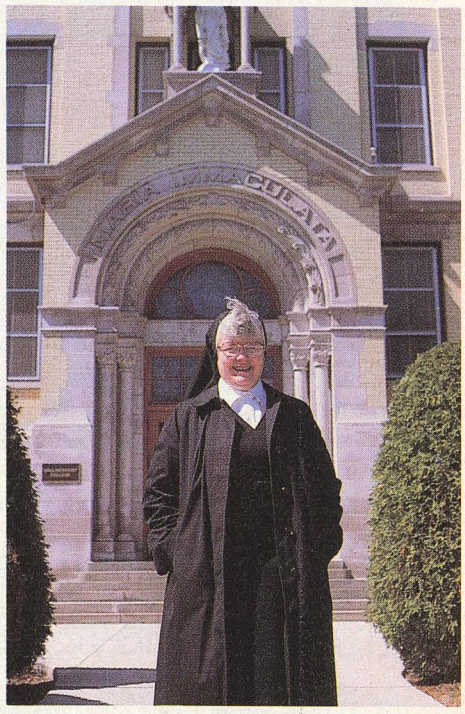
TECMAR
Tecmar Sweetens Apples™

Tecmar Inc. 6225 Cochran Road Solon (Cleveland), Ohio 44139-3377 Phone:(216)349-0600 Telex: 466692









Advertisers

Apple's® new baby has

Microsoft BASIC
on Apple's new Macintosh

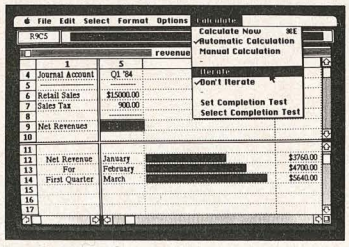
It's called Macintosh™. And it has our brains and a lot of our personality.

We're called Microsoft®. And our part of Macintosh is five new programs that are bright, intuitive, outgoing, understanding and born to perform.

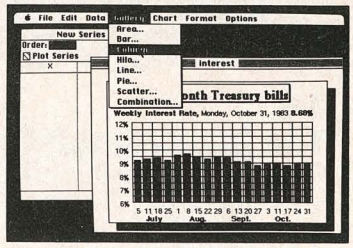
Our pride, your joy.

Taking advantage of Macintosh's mouse and rich graphics, we've designed software that works like you, even thinks like you.

All our programs share the same plain English commands. So what once took days to learn, now takes hours or minutes to learn with Macintosh.



Microsoft Multiplan



Microsoft Chart

Meet the family.

Our financial whiz is MULTIPLAN®, an electronic spreadsheet that actually remembers how you work. Even offers suggestions on spreadsheet set-up.

When it comes to writing, nothing travels faster

our best features.

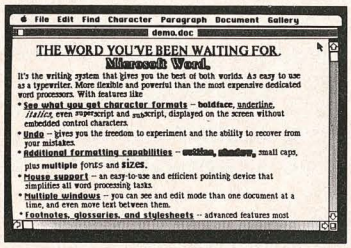
than our WORD. Using the mouse, it lets you select commands faster than you can say "cheese."

Our most artistic child is CHART. It gives you 40 presentation-quality chart and graphic styles to choose from.

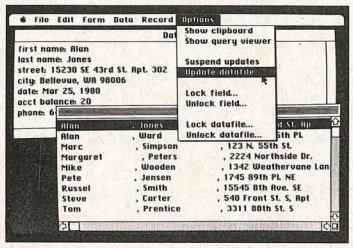
FILE is our most manageable child, an advanced personal record management program.

MICROSOFT®
The High Performance Software

And BASIC, the language spoken by nine out of ten microcomputers worldwide, is the granddaddy of them all. Now enhanced to take advantage of the Macintosh mouse, windows and graphics.



Microsoft Word



Microsoft File

We'll be adding more to the family soon. So call 800-426-9400 (in Washington State call 206-828-8088) for the name of your nearest Microsoft dealer.

ST.MAC

Chairman	John Haller
President	Margot Comstock Tommervik
Vice President	Al Tommervik
Publisher	Mary Sue Rennells
Editor	Patricia Ryall
Art Director	Kurt Wahlner
Managing Editor	Catherine Petersen
Editorial	
Associate Editors	David Durkee
	Michael Ferris
	David Hunter
	Matt Yuen
Contributing Editors	Jean Varven
	Alden Woodard
Special Assignments	Todd Zilbert
Copy Editor	Cordell Cooper
Assistant Copy Editor	Judith Pfeffer
Proofreaders	Harry McNeil
	Steve Thomsen
	Marjorie Kaufman
Editorial Assistant	Betsy Barnes
Art	
Assistant Art Director	Lucas McClure
Production Manager	Donald J. Robertson
Asst. Production Manager	Nancy Baldwin
Ad Production	Michael G. Pender
Assistant	Ruth Seid
Business	
Controller	Duane E. Runyon
Vice President of Finance	Charibel R. Hilario
Advance Projects	Steve Shendelman
Accounting Assistant	MaryAnn Nail
Advertising	
Coordinators	Linda McGuire Carter
	Cathy Stewart
West Coast Sales	Mike Antich
	ST.Mac
	7250 Laurel Canyon Boulevard
	Box 7041
	North Hollywood
	CA 91605
	(818) 980-5074
East Coast Sales	Ian Ross
	Paul McGinnis
	Advertising Sales
	690 Broadway
	Massapequa, NY 11758
	(212) 490-1021
Midwest and Rocky Mountain Sales	Ted Rickard
	Kevin Sullivan
	Market/Media Associates
	435 Locust Road
	Wilmette, IL 60091
	(312) 251-2541
Circulation	
Trial Subscriptions	Deirdre Galen
	Anna Gusland
	Ramona Gordon
Paid Subscriptions	Michelle Vigneault-Kirschenbaum
	Janeth Godoy-Aguiar
	Barbara Naimoli
	Josie Walley
	Laurie Ure
	Jill Stroud
	Nancy Kelly
Back Issues	Gene O'Day
Dealer Sales	Lashea Lowe
	Leticia Garcia
Systems	John Heitmann

Credits: Composition by Graphic Typesetting Service, Los Angeles, California. Printing by Storm Printers, Memphis, Tennessee.

Apple, Lisa, and Macintosh are registered trademarks of Apple Computer, Inc., Cupertino, California. Softalk is a trademark of Softalk Publishing Inc., North Hollywood, California.

ST.Mac. Volume 1, Number 5. *ST.Mac* is published monthly by Softalk Publishing Inc., 7250 Laurel Canyon Boulevard, North Hollywood, CA 91605; telephone (818) 980-5074. Second-class postage pending at North Hollywood, California, and additional mailing offices.

Postmaster: Send address changes to ST.Mac, Box 7041, North Hollywood, CA 91605.

Free Subscriptions: Complimentary trial subscription to all owners of Lisa and Macintosh computers in the U.S. and Canada. If you own a Lisa or Macintosh, send your name, address, and the serial number of your machine to ST.Mac Circulation, Box 7041, North Hollywood, CA 91605. Please allow six to eight weeks for processing. *ST.Mac* is totally independent of Apple Computer, Inc.

Paid Subscriptions: $24 per year. Special rates for schools and libraries, $12. Concurrent additional subscriptions for schools and libraries, $8 each. Please allow six to eight weeks for processing.

Back Issues: $3. Please allow six to eight weeks for processing.

Problems? If you haven't received your *ST.Mac* by the fifteenth of the month, or if you have other problems with your subscription, Michelle Vigneault-Kirschenbaum can help out. Call (818) 980-5074 or (800) 821-6231.

Moving? Send new address and a recent label from *ST.Mac* to ST.Mac Circulation, Box 7041, North Hollywood, CA 91605; telephone (818) 980-5074 or (800) 821-6231. Please allow six to eight weeks for processing.

VIEWPOINT

Gucci Shoes and Macintosh: the Surprise of Quality

There's a time when it's premature to say a product is successful. And there's a legitimate moment to announce the success of a new machine.

Let's do it now: The Macintosh is a success. Macs are being gobbled up like popcorn all across the country. Reviews of the machine have been stellar. Businesspeople are crunching spreadsheets, and hobbyists are using Microsoft Basic to create marvels that only hint at what's to come (when Apple's MacBasic finally arrives). Apple Inc. says the company will move 350,000 machines this year, and it probably will—if it can make that many. Dealers think Macintosh is the greatest thing since, well, that other machine.

Life is good.

But wait a minute. Why *is* Macintosh successful, anyway?

Maybe despite itself. It's possible, in fact, to build a strong argument *against* Macintosh: not enough memory, not enough software, not enough—well, those two make a good case by themselves.

Then there's the phantom "external disk drive" promised by Apple. And the fact that plain old IIe's are priced low enough (now) to tempt all but the most die-hard Macophiles. The IIc looks pretty good, too.

And 2,500-odd dollars is real money, right? And *1-2-3* isn't even on the horizon.

So whence the success? Is it the size? The nifty stuff in ROM? The speed of the machine? The fact that text looks like, aah, text? The mouse (though it didn't work for Lisa, did it)? Maybe just a lucky combination of those traits and more? We'd like to know. You'd probably like to know. The other computer manufacturers would *love* to know!

Here's one guess. It has to do with the vision of Steve Jobs and goes all the way back to the fabled Apple II.

As the story is told, the original Apple II was preceded by, not surprisingly, the Apple I. The Apple I was a real computer, all right, with all the normal real computer stuff. But there was one thing it didn't have. (No, not a heart, or a diploma, or a proclamation. That's a different story.) What the Apple I didn't have was a case.

A case! Not a carrying case, but a case for the computer—the plastic stuff within which the actual computer part of the computer exists. No case.

In those days, it wasn't a big deal not to have a case. But Steve Jobs wanted a case. And got one. But the point (finally, the point!) wasn't the case. The point was the idea of "products." Not computers, *products*. Good consumer products. Which just happened, in this instance, to be computers.

One of Jobs's favorite metaphors for all this is the Cuisinart. You know: the great French food processor. The Cuisinart, if you've ever used one, is a good product; it has a great dependable motor, is utilitarian, works like a breeze, and the cord's not too short. A good product, all around, that just happens to be a food processor. You may prefer knives, but you get the idea.

More good products? How about Ferraris? Gucci shoes. Dresden china. Harris tweed jackets. Smithfield hams. Nautilus exercise machines. *National Geographic* magazine. We could go on for a long time with this: small products and big products and expensive products and inexpensive products, but all quality products.

That's all you need to know to understand Steve Jobs in particular and Apple Computer in general. A commitment to good products that, in this instance, are computers.

And that may just possibly be it. Macintosh is a good product first and a computer second. Maybe the first Gucci shoe this industry has ever created. It's not any one thing, and it's not even the combination of everything. It's a totality, a gestalt. An attitude that dots all the i's, rounds all the corners, and cares as deeply about what's on the box as what's in the case.

The point is not to make it only as good as it needs to be, but to make it better than anyone would even *expect* it to be. The surprise of quality. A risky gamble in a cutthroat marketplace that twirls on features and price and "Will it run *1-2-3*?"

But, of course, that may *not* be the reason behind the success of Macintosh. That's just one case, and maybe not the right case.

But sometimes when you announce a success, whether it's a marriage or a memo or a good night's sleep, you just have a feeling that you know the reason.

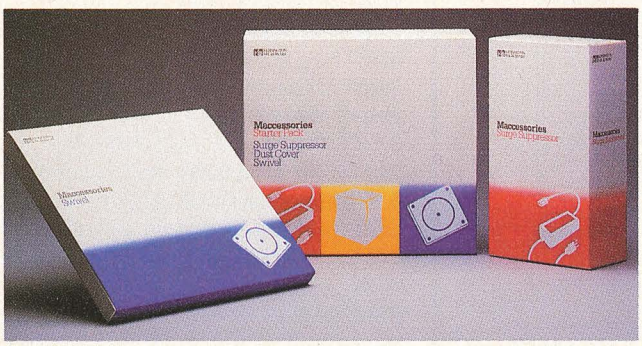
That's how we feel this month. And we're guessing you feel the same way.—A.W.

Maccessories!™

The Maccessories SWIVEL
is a light weight, durable lazy susan that fits neatly under your Macintosh. The SWIVEL revolves a full 360° and lets you adjust the viewing angle of your Macintosh™ with the touch of a finger.

Since the SWIVEL attaches to the bottom of your Macintosh computer, it goes everywhere your Macintosh goes, even inside the Macintosh carrying case.

Maccessories STARTER PACK.
Start your Macintosh off right with this money-saving kit containing the Maccessories SWIVEL, SURGE SUPPRESSOR *plus* an anti-static DUST COVER to keep your Macintosh clean and dust-free.



The Maccessories SURGE SUPPRESSOR
protects your Macintosh against power problems. Its circuitry stops surges and spikes, and filters out aggravating line noise. Simply replace your Macintosh power cord with the Maccessories SURGE SUPPRESSOR and you're ready to go.

Now also available:

Maccessories (PORTABLE) MODEM
Operates at 300 bps with any communications software. It's inexpensive, rugged and easy to use. Warranted for 5 years.

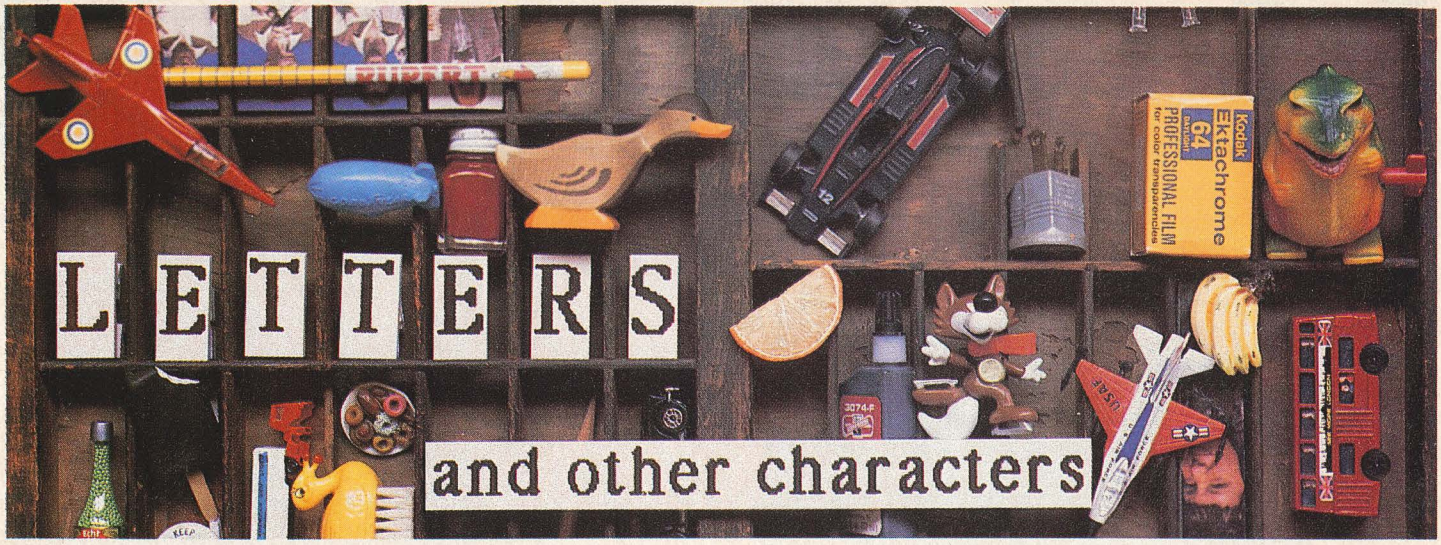
It's small (7½" x 3½" x 1½"), powered by a 9-volt battery, less than 1 lb. in weight, and slips easily into the Macintosh carrying case.

Maccessories DISK CASE
Provides safe storage for 36 Macintosh disks. Five dividers help you organize your disks. Comes with 10 spare disk labels.

Available at Apple® dealers everywhere.

KENSINGTON™ MICROWARE

Kensington Microware Ltd. 251 Park Avenue South, New York, N.Y. 10010 (212) 475-5200 Telex: 467383 KML NY



Basic COMments

In the April issue, you claim that Microsoft Basic lacks a COM statement [Miscellanea: Where's the COM Statement?]. In fact, it does let you open COM1:, just like on the IBM PC, but it lacks the ability to set the baud rate, parity, and so on with this statement. However, with a little machine language subroutine magic, you can do lots of things. Witness Dennis Brothers's program called *MacTEP*, which is available on the MicroNet Apple Users Group on CompuServe (go pcs-51). The program is now up to version 1.81, and Dennis is threatening to give us xmodem and CompuServe protocol file transfers. It's pretty good considering MS-Basic is all we have to work with!

Dave Alverson
Cincinnati, OH

The People's Computer

I just received my Macintosh through the Own-A-Mac program for Apple dealers, as well as the April issue of *ST.Mac*. I found your magazine very entertaining and read it from cover to cover. Although most magazines stress the unique features of the Macintosh, *ST.Mac* seems to be more oriented to how those features translate to the user.

Having owned a Lisa for about six months now, I can honestly state that the Lisa and the Macintosh are the only two computers with which one concentrates on the work being done, *not how to do it*, regardless of the application being used! Additionally, they open up a totally new form of graphic expression heretofore unavailable on any other personal computer, and it is within everyone's reach, not just that of the hobbyists.

MacPaint wins hands down as the most addictive program I have laid my hands on, although my overall preference goes to *LisaDraw* and *MacDraw*. People complain about Macintosh's 128K memory; however, if they insist on having a 512K Mac and are not willing to wait, why don't they purchase a Lisa 2? Lisa owners give up portability, but with MacWorks, Lisa emulates a 512K Macintosh, plus it has the potential of running the Lisa Office System software (with the addition of 512K of memory and a hard disk) or the next standard in operating systems: Bell Labs's multiuser Unix (System III or System V, with the addition of a hard disk). All are available *now*!

George J. Jahchan
Montreal, Quebec, Canada

Just Fooling Around

Congratulations on one of the funniest April Fool's pieces in any magazine. The Allen Munro/Alan Kay interview ["Alan Kay Thinks...."] ranks right along with BBC-TV's famous "Spaghetti Tree" episode of 1957.

Alan Kay was kidding, wasn't he? Of course he was! Allen and Alan are in a bar somewhere right now laughing about the gullible fools who got upset that *ST.Mac* printed the PARC priesthood story.

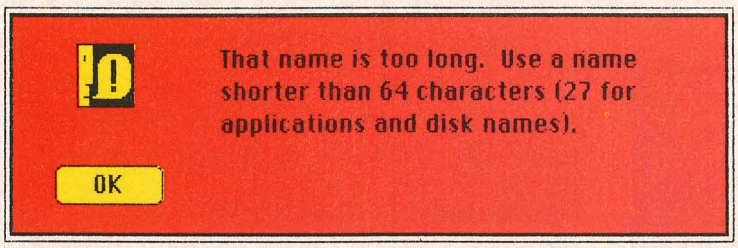
It *was* a put-on, wasn't it, fellows? Allen? Alan? Hey, guys....

Barry W. Collins
Demopolis, AL

Thanks for a fantastic April issue and for a perfect interview with Alan Kay. I wish Kay good luck, but he should consider this: Parents are generally too busy to get as involved as we would all like in their kids' education. Some parents aren't really interested—and although it's common knowledge that many parents want their children to become the people they never were, they also do not want to be surpassed *too* much by their offspring. It's a situation in which two-thirds of the key people (parents and teachers) are not only uninformed but also don't want to be informed. At fifteen, I am fortunate enough to have parents who do.

The Erroneous Zone

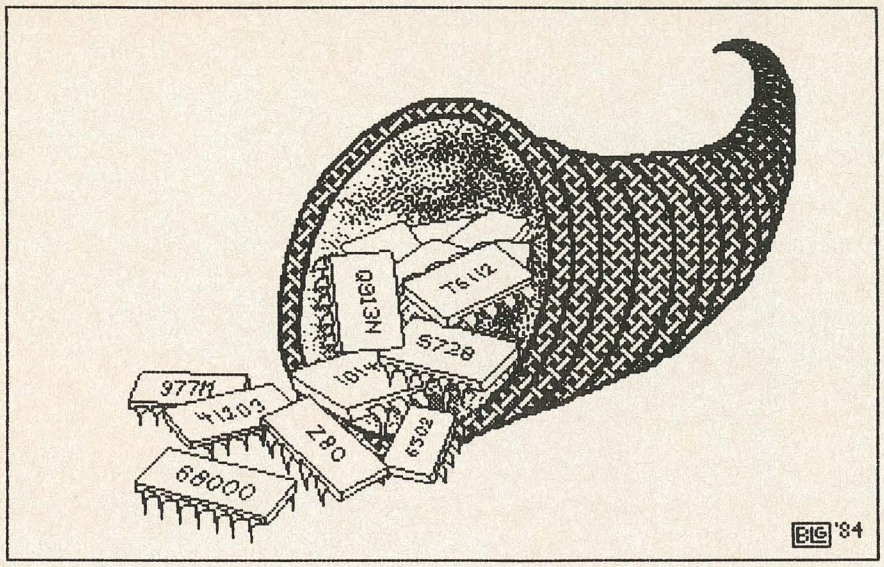
I really appreciate your magazine, but I have found an error in your April issue [Viewpoint: Making Mac PC-Compatible]. True, it isn't a big error, but it is something that I think should be corrected. In this column, A. Woodard states, "Mac's Finder also has limitations on filenames: They can't be longer than 255 characters." This is clearly erroneous. If one tries to enter a long filename, one is greeted on the screen by...



Other than that, I didn't find any glaring mistakes, but you'd better keep on your toes because I'll keep looking.

Dan Eldridge
Larimore, ND

Umm...Woodard managed to squirm out of this one with a feeble but dignified explanation recounted in last month's Miscellanea: "It seems that the Macintosh operating system software (held in ROM) does indeed provide for filenames of 255 letters.... The filename includes both the drive name and the name of the filename. Thus, the filename 'This is my favorite MacWrite disk:And this is my favorite file on the MacWrite disk' is a valid filename. The words to the left of the colon identify the disk, and the words to the right of the colon identify the file."



About Microsoft Basic for Macintosh: I urge no one to buy it, as it is the shoddiest version from Microsoft I have ever seen. I would rather program in Applesoft on the Apple II. For example, the LINE command, for drawing a line, will not accept more than two coordinate pairs; in order to draw a number of connected line segments, you must repeat the command for every point after the second.

I took you up on your offer and sent in your card for a free subscription. In return, enclosed is a free cartoon.

Ben Geer
Miami, FL

A Fatal Mistake

I had just completed interconnecting the Macintosh to my venerable Apple II. Soon a vast library of Applesoft public domain software would be uploaded to the Mac.

With a great deal of anticipation, and an equal amount of foreboding, I loaded a terminal program into the Mac (Dennis Brothers's *MacTEP*, available for downloading from The Source, CompuServe, and numerous local bulletin boards). So far, so good. It asked me for the baud rate and the name of the file I wished to save. I quickly typed the necessary information.

All was in readiness for the next crucial step, the transferring of a Basic program from the Apple to the Mac. I very slowly typed PR#2. This step would activate the serial card in slot 2. I then gave the Applesoft command LIST. My eyes were riveted to the Mac screen. If all went well, the program in the Apple would be sailing into the Mac at flank speed.

I waited and waited and...nothing. Was it a hardware or software problem? Was it in the Apple or the Macintosh? I rapidly checked everything. Had I misplaced a comma in the software I had so carefully transcribed for the terminal program? No, it seemed to be okay. Was there a loose connection in the cable I had fabricated for this project? No, that checked out. The baud rate in the Mac software was 9600...check. The DIP switches on the serial board were Off, Off, On. Wait a second...there it was! They should have been set at On, On, Off.

With a great deal of optimism I was ready to try again. This time the program listing flew across the Macintosh screen. Success! There it was, my Applesoft program. Oh, it wasn't perfect. There were some minor problems with carriage returns and forty-column listings versus eighty-column screens, but these were minuscule in comparison with my historic achievement.

Just then my wife announced that her friend had arrived to print some mailing labels on the old reliable Apple. No problem. I had just climbed Mount Everest and wished to savor my accomplishment. All that had to be done to get the Apple ready for the mailing lists was to get all the cables put away and remove the serial card. I carefully turned off the Macintosh and pulled out the serial card.

Oh, no! I had turned off the *wrong* computer. The serial card had departed Earth at the speed of light.

Oh, well, what's a few bucks for some new chips after what I had succeeded in doing?

Larry Margulis
Cherry Hill, NJ

Making Book on Mac

I'm writing a book on art and the Mac and would like to speak with people who are using the graphic arts capabilities of the Mac in their trade or business. This would include artists, planners, architects, designers, printers, illustrators, technicians, publishers, art directors, and so on. Please call or write me at 2139 Newcastle Avenue, Cardiff by the Sea, CA 92007; (619) 942-3838.

Vahe Guzelimian
Cardiff by the Sea, CA

ST.Mac would like to hear your reactions to articles as well as your views about the world of personal computing. Send your opinions, queries, suggestions, and solutions to ST.Mac, Box 7041, North Hollywood, CA 91605.

THE SEDUCTION OF THE INDEPENDENT DEVELOPER

Apple's Aggressive Campaign to Market Mac Software

By Guy Kawasaki

It's no secret that Apple wants the Macintosh to follow in the footsteps of its first successful computer, the Apple II. The Apple II has a track record that no other personal computer can match. It has been on the market for nearly eight years and has a software base exceeding ten thousand commercial programs.

But the computer market is a different place now than when the Apple II was introduced. A lot of the original Apple II programs were created by ordinary users hacking in their living rooms into the wee hours of the night. Many of these programs were written in Integer Basic and Applesoft, the versions of Basic that were included in the ROMs of the Apple II and the Apple II Plus. For a program to be accepted by today's more demanding consumers, it must be fast, slick, and powerful. And marketing considerations frequently demand that it be written in assembly language.

This is especially true of software for the Macintosh and the Lisa, which have no language built into their ROMs, and which have perhaps the highest standards for software designers to meet in the industry. Apple hasn't put its resources into stocking the shelves with software for the Macintosh and Lisa. Instead, it has been actively seeking and supporting companies that it believes will become the pioneers of a whole new software explosion to come.

Let's look at the process of becoming a Macintosh software developer.

Guy Kawasaki is the head of software evangelism and technical support for Apple's Macintosh and Lisa products.

How To Get Started

In order to develop Macintosh software, a developer needs documentation, equipment, and software. The suggested development system currently includes documentation (*Inside Macintosh*), equipment (Macintosh, external disk drive, Imagewriter printer, Lisa 2/5, Lisa half-megabyte RAM card, and Lisa Imagewriter Accessory Kit), and software (Lisa Pascal Workshop and *Macintosh Software Supplement*). Let's take a closer look at each of these components.

Documentation

The documentation of Macintosh software development is called *Inside Macintosh*. It is a two-volume set of photocopied manuscripts containing detailed explanations of the Macintosh ROM, RAM-based packages, user interface guidelines, and the Lisa development environment. *Inside Macintosh* shows a developer how to use the software routines developed by Apple to implement features like high-speed graphics, windows, and pull-down menus as they appear in *MacPaint* or *MacWrite*.

Inside Macintosh presently concentrates on software development; it includes only limited hardware documentation. The final version of *Inside Macintosh* will have more information on both hardware and software. It will be available as a finished, published manual in the fourth quarter of 1984 in Apple dealerships and bookstores. Any interested party may purchase the current version from the Apple mailing facility listed at the end of this article.

Equipment

Serious commercial development of Macintosh software requires a Lisa at this time; while standalone languages will be available in the coming months, they are not yet on the market. Until we get Mac-based development tools, programs must be written on a Lisa, compiled, linked, and then downloaded to the Macintosh. When the necessary software arrives, the Macintosh will be able to act as its own development system, without the aid of a Lisa 2.

Additional equipment is necessary for various reasons. The extra half megabyte of RAM, which brings the Lisa 2/5 up to a one-megabyte capacity, is required to run the Lisa Workshop development software. The second disk drive for the Mac enables a developer to produce backup copies more quickly and to create software that will use a second drive. The Imagewriter printer is used to generate listings and to test printing routines, and the Lisa Imagewriter Accessory Kit provides the cable for printing source code from the Lisa.

The development equipment is available from local Apple dealers or, for qualified developers, through the Apple Certified Developer Program.

Software

Most developers currently are using the Lisa Workshop with Lisa Pascal for Macintosh program development. To augment that, C for the Lisa Workshop will be available from Apple in July. The Workshop and languages are powerful development environments, combining an editor, assembler, optimizing compiler, debugger, and linker.

Though other compilers exist for 68000 machines, the Workshop provides a full set of utilities and software interfaces to the Macintosh ROM. These capabilities are essential for implementing a true Macintosh user interface. By using the Apple ROM routines, a developer saves time, avoids making end users learn a new interface, and conserves RAM and disk space.

Beginning this summer, Apple will release three standalone development languages: MacAssembler/Debugger in July and Pascal and C in December. These packages will contain full interfaces with the Macintosh ROM and will allow the programming of commercial applications. When these languages become available, it will no longer be necessary for Macintosh developers to purchase the Lisa 2.

Those wanting to write programs purely for their own use can do so now with Microsoft Basic. Soon, Macintosh Basic and Macintosh Pascal will be available as well. However, these languages are not intended for the development of commercial applications like word processors and spreadsheets. The only language currently available for commercial, standalone development is MacForth, from Creative Solutions in Rockville, Maryland.

The final software that a professional developer needs is the *Macintosh Software Supplement*. The *Supplement* contains Macintosh libraries, utilities, and sample programs. MacWorks, the Macintosh emulation software for Lisa, is also provided so that developers can test their software for Lisa compatibility. Purchase of the *Supplement* entitles a developer to updates to the *Supplement* software and *Inside Macintosh* and a copy of the first released version of *Inside Macintosh*.

The Apple development languages are available through Apple dealers or, for qualified developers, through the Apple Certified Developer Program. The *Software Supplement* is not sold in dealerships at present; it is available to any interested party from the Apple mailing facility.

Assistance from Apple

The framework for the overall relationship between Apple and the third-party developers is provided by the Apple Certified Developer Program. A developer who has been certified by Apple can attend conferences and seminars, purchase equipment and software at a discount, and apply for technical support. The program is administered by Software Industry Relations, a part of Apple's corporate marketing department.

To qualify for certification, a developer must provide background information about the company and demonstrate its ability to bring Apple-compatible products to market. Applications are screened by Software Industry Relations, which makes a decision based on such factors as a company's business plans, existing retail products, technical expertise, and marketing experience. The program is intended for developers and publishers whose products will be sold in retail and computer stores. Certified Developer applications are available from Software Industry Relations at Apple Computer Inc.

The Macintosh Developers Group, a part of the Macintosh division marketing department, provides Macintosh-specific development, as well as design and marketing assistance. Activities of this group include software evangelism, technical support, MacColleges, and marketing programs.

Software evangelism, as Apple calls it, is the function of recruiting companies for Macintosh development. Prior to Macintosh's release, Apple's representatives convinced more than 100 companies to develop products for Macintosh. Since its release, they have provided marketing, technical, and procedural information to more than 2,500 developers.

Technical support is available through the Macintosh Registered Developer Program. For a $500 fee, the developer purchases from Apple a "package" of services for a six-month period that includes extensive telephone support and consultation. Registered developers receive assistance with user interface design and techniques for implementing features such as windows, text editing, dialog boxes, printing, and pull-down menus. After the six-month period, developers usually require little additional support.

To qualify for this program, a developer must be an Apple Certified Developer and demonstrate a serious commitment to the development of commercial products. Registered developers are expected to assign their top-flight programmers to the project and develop products that adhere to the Macintosh user interface guidelines. A brief application is completed by the developer and reviewed by the Macintosh Developers Group. The Registered Developer Program is for serious commercial developers of products and not end users.

Another kind of technical support is a series of training seminars called MacColleges; their purpose is to accelerate the completion of products by experienced Macintosh developers. Programmers who have been working a minimum of two to three months can bring their source code to MacCollege in Cupertino, California, for three days of lectures, lab work, debugging, and consultation with Apple's technical people. Only developers with partially completed applications are admitted because the lectures and debugging sessions are for advanced programmers. MacCollege attendees must be Apple Certified Developers.

The MacCollege sessions cost developers $500. The lectures are held in the morning and private debugging sessions in the afternoon. The sessions are conducted in a facility called the Macintosh Software Mill, which is fully equipped with a lecture area and fifteen

MacTote.™

The highest quality case built specifically to carry your Macintosh™ Computer.

Now your Macintosh Computer can leave your home or office fully protected and in style.

The **MacTote** computer case is clearly the finest carrying case manufactured especially for your Macintosh Computer, and for a number of easy-to-see reasons:

1. Padded pouch for mouse; 2. Open pouch for manuals or notes; 3. Open, padded pouch for keyboard; 4. Side handle for easy carrying and stability; 5. Padded, Velcro handle (no snaps); 6. Padded, non-slip shoulder strap; 7. Zippered pouches for disks or accessories on both sides of case; 8. Zippered pouch for accessories, modem, cords, etc.; 9. Padded pouch for disk drive, and extra padding to protect Macintosh's screen; 10. Reinforced bottom fully supports and distributes weight of Macintosh; 11. Heavy-duty metal zippers.

Optimum Computer Luggage has extensive experience in making top-of-the-line totes and accessories and **MacTote** is the latest in a long line of beautiful and functional luggage. Made of long-wearing and completely water-resistant Cordura, **MacTote** gives you every feature you'll ever need for carrying your Macintosh and peripherals.

Another attractive feature is **MacTote's** low price, which makes it even easier to pick up and carry away.

Call today for the dealer nearest you so you can get a **MacTote** for your Macintosh.

Who says you can't take it with you?

Call (619)287-2886
Or in CA TOLL FREE (800)632-4200.

OPTIMUM
COMPUTER LUGGAGE

The Finest Quality Doesn't Have To Cost More.

4445 Fiftieth Street
San Diego, CA 92115

private cubicles containing Macintosh/Lisa development systems.

MacColleges are conducted in Cupertino so that Apple's engineers can assist developers in their particular areas of expertise. The engineer that wrote Apple's printer driver, for example, may help a developer with printer driver questions. This kind of intensive support is nearly impossible with regional seminars.

For further information about the Macintosh Registered Developer Program and MacCollege, contact Software Industry Relations at Apple.

Support for the Little Guy

Recognizing the need for introductory technical support for beginning Macintosh developers, Apple has trained third-party organizations to conduct regional seminars. One such company is the Technology Services Group of Bellevue, Washington. The Technology Services Group conducts seminars for companies that are starting or considering Macintosh development.

Marketing Goodies

The Macintosh Developers Group helps developers bring Macintosh products to market in three ways. First, developers can receive a promotional listing in a monthly publication called the *Apple 32 Development Team List*. This list is printed in quantities of five thousand and distributed to dealers, sales representatives, publications, and analysts.

Second, developers can familiarize Apple dealer sales personnel with their products by participating in Apple's Own-A-Mac program. The Own-A-Mac program allows Apple dealer sales personnel to buy Macintoshes at extremely low prices for personal use. Developers can participate in this program by offering their products to Own-A-Mac salespeople for at least 60 percent below retail. This introduces their product to the people who will be selling it.

Finally, developers can have their sales promotional material and demonstration software distributed to the twenty-five hundred Apple dealers and sales offices. The Macintosh Goodie Box, as this program is called, is

To purchase *Inside Macintosh* ($150) and the *Macintosh Software Supplement* ($100), send payment in advance to:
Apple Computer Inc.
467 Sarasota Avenue, Suite 621
Cupertino, CA 95014
(California developers add sales tax)

For information about the Apple Certified Developer Program, the Macintosh Registered Developer Program, MacColleges, and *Apple 32: Developer's Handbook*, contact:
Software Industry Relations
Apple Computer Inc.
20525 Mariani Avenue, Mail Stop 23AF
Cupertino, CA 95014

conducted several times a year, and developers can submit their materials for possible inclusion in the Goodie Box. For further information about these marketing programs, contact the Macintosh Developers Group at Apple.

Advance to Go

Macintosh development requires a significant allocation of time and resources, and Apple suggests that developers follow a three-step procedure to ensure that they're making the correct decision.

First, they should obtain a copy of the *Apple 32: Developer's Handbook* and an application to the Apple Certified Developer Program. The *Developer's Handbook* contains information about Apple's marketing plans, products and accessories, development paths and environments, and startup procedures. Both the *Developer's Handbook* and the application are available from Software Industry Relations.

Second, developers should complete the Certified Developer application and purchase a copy of *Inside Macintosh*, the documentation manual. By reading *Inside Macintosh*, a developer can assess the technical requirements of producing a Macintosh application.

Third, developers should purchase a development system including the hardware and software outlined in this article.

Macintosh provides an excellent opportunity for software developers to profit from the introduction of a new personal computer from Apple. Apple intends to help developers take advantage of this opportunity by providing design, technical, and marketing support.

Apple's goal is to make Macintosh the next personal computer standard with the help of its partners, the independent developers.

Wooing Developers in the Big Apple

The first step in getting a large body of Macintosh software to market has been to woo independent developers. To this end, Apple has sponsored a series of one-day workshops for the decision-makers of potential Mac software publishing companies. From February through April, seminars were held in Boston, Chicago, Dallas, New York, Santa Clara, and Anaheim, California. Abbreviated presentations were made at the West Coast Computer Faire and Softcon. Kevin Goldstein attended the last of these conferences and filed this report for ST.Mac.

With the completion of the April Macintosh Developer's Workshop in New York, Apple Computer has closed down the first phase of an ambitious plan aimed first at enticing commercial software developers to write programs for the Mac and then at expediting those programming efforts.

Though its billing as a "Developer's Workshop" may seem to imply that the one-day seminar was chiefly an introduction to the technical esoterica involved in writing programs for the Mac, Apple took the opportunity to convince its audience of the inherent advantages of writing Mac software.

While this tack may have been necessary when Mac was introduced four months ago, the 50-50 split between technical and marketing talk seemed unnecessary at this last conference—almost all of the hundred-plus attendees appeared to have already made up their minds in favor of Mac. That impression was strengthened at the conclusion of each of the bravura technical talks presented by Alain Rossmann, when the audience broke into spontaneous applause. This was a group that wanted the cake, not the icing.

And Apple seems intent on delivering just that. The company is going after software developers with the kind of gusto usually seen only in beer commercials. Recognizing that the success of the Mac depends not only on getting a lot of good software onto dealers' shelves but also on getting it there fast, Apple is targeting commercial software developers for special treatment. That treatment currently consists of a combination of healthy discounts on Macintosh development systems plus the kind of technical support IBM PCers can only dream about.

That kind of support is expensive. The discounts probably don't hurt Apple much, but tech support doesn't just eat up manpower, it eats up manpower that's working on future Macintosh products—and Apple intends to lavish its support on only those developers it believes have the ability and the determination to develop and publish (or to have published) professional quality software. To keep the kibitzers out, the company is restricting entrance to both the Certified Developer and Registered Developer programs with some pretty rigid requirements. Those requirements all have just one point: to prove to Apple that the applicant has "the right stuff." And if you look at the situation from Apple's perspective, it's possible to figure out just what, in this case, constitutes the right stuff.

Apple wants programs for Mac, and it wants good programs. Despite a lengthy application form, all an applicant really has to do is prove to Apple's satisfaction that it can and will produce and publish some quality software in a reasonable amount of time. That means an applicant with a product already on the market is practically a shoo-in; lacking that, a good brochure will help immensely.

Apple is asking startup companies for a detailed business plan, and when Apple says detailed, it means *detailed*—thirty to forty pages is considered appropriate. If you're wondering how anybody at Apple is ever going to have the time or inclination to read such a monster, be assured that at the workshop, developer relations program manager Dennis Gallinat swore that all those plans were being read. In view of the fact that Gallinat says that more than four thousand applications had been received as of April, somebody at Apple must be doing an awful lot of reading. Just for the record, about 70 to 80 percent of the applicants are being accepted.

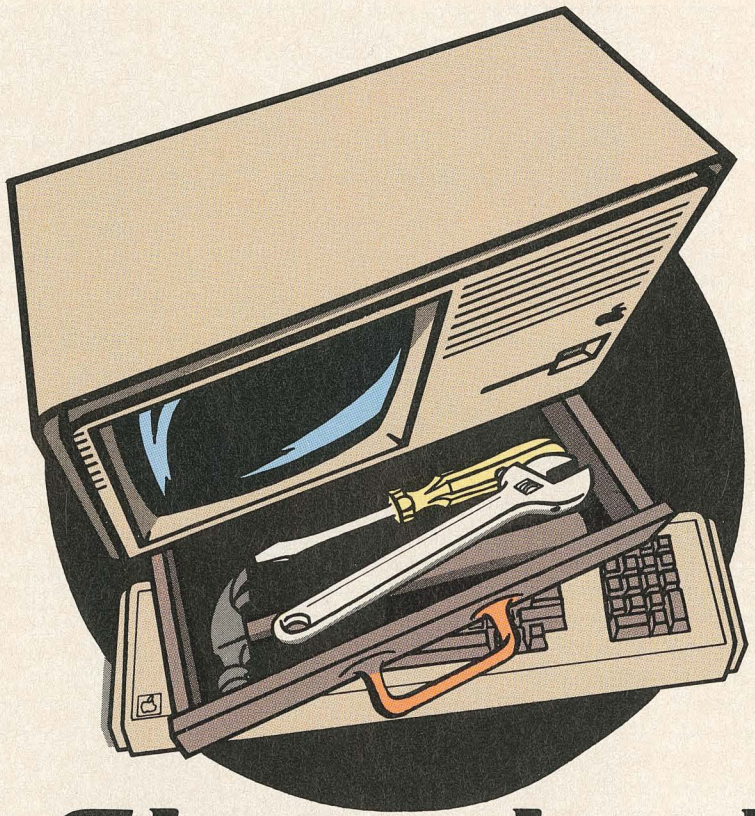
However good an applicant's credentials are, luck might still play a role when it comes time for the application to be approved. Apple is trying to get a broad mix of software, which means that the application from the second company planning to write a spreadsheet is less likely to be approved than the first company's, and the third is even less likely than the second.

Gallinat announced a final step in the development chain called MacCollege. Limited to developers with a complete or nearly complete product, the program is an intensive three-day affair, consisting of workshops and lectures, that aims at fine-tuning and applying the finishing touches on about-to-be-released programs. MacCollege will be held once a month in Cupertino, at its own digs next to the Mac development group. If anybody runs into a problem that the instructors can't solve, some pretty powerful talent can be summoned from just around the corner.

What is needed to get the wheels of development moving is a free flow of information, and Apple, having not totally forgotten its origins, will likely cooperate with third parties to establish courses open to the public. In the meantime, it was decided at the workshop that more information of a technical type will be made available through *ST.Mac* and *Softalk*. (Okay, and *Macworld* too.)

While Apple's seduction of the commercial developer is a unique opportunity for those qualified enough to take advantage of it, it sets up a situation wherein the garage shop tinkerer is likely to feel neglected. The problem is relative; IBM, for example, does nothing for anybody. which means that everybody is equally neglected. (It's more democratic, if nothing else.) In any case, many of the perks currently offered under Apple's programs wouldn't do the small developer any good at this time anyway, since commercial development can't be done on a Macintosh-only system—a Lisa and a Mac will be required until some standalone software makes its debut later this year.—Kevin Goldstein

The Lisa Toolkit



In a Class by Itself

By David Durkee

Ever since computers were first invented, computer languages have been used to communicate with them. From the fast, low-level machine languages—the complex numeric codes used to give instructions directly to the central processing unit—to the more human-oriented, higher-level languages that the majority of programmers use, computer languages have been our primary means of talking to the machine. Even the end user, who may never have wanted to write a computer program, has had to learn at least enough of the computer's language to tell it what programs to run.

Lisa Technology has eliminated the need for end users to learn computerspeak to communicate with Apple's thirty-two-bit machines. The methods a user must learn to run a program on Lisa combine the simplicity of pointing a finger with the clarity of a universally understood set of symbols. There are a few special words to learn, but certainly not enough to be called a language.

This degree of simplicity has yet to reach the programmer, however. In order to tell Lisa how to do all the complicated actions that make up something like *LisaDraw* or *LisaProject*, the programmer still has to be able to use special languages.

The Language Is the Program

A Pascal-derived language called Clascal, together with Toolkit, a package of routines written in Clascal, is the basis for the development of applications programs for the Lisa 2. Clascal is more than just a new language; it embodies a new concept of computer programming that experts are getting very excited about—the object-oriented language.

Until now, all computer languages have been oriented toward procedures. The syntax consisted entirely of commands that told the computer what to do. Pascal, a popular language in the Apple II world, is notable because it imposes an organized structure on program procedures. Once a procedure has been defined, applying it to many different instances is relatively easy; the programmer merely has to say *do this procedure* and pass along the necessary details—the data that specifies how the procedure is to be performed. Such things, before Pascal, were often haphazard.

Besides organized procedures, another important aspect of human language was never clearly expressed in computer language. That is, that actions involve objects. Clascal allows the definition and manipulation of objects.

Objects were only implied, however, in earlier computer languages. They were created or acted upon by the procedures, but never overtly defined, leaving their meaning nebulous. Clascal breaks with this established way of doing things. It was developed especially for the Lisa and does for objects what Pascal does for procedures. Clascal recognizes, and forces the programmer to note consciously, the properties of objects. Objects in Clascal, like objects in the world at large, are defined by classes and exceptions. One object in Clascal acts as an ancestor to all the others. For instance, if "dog" were the ancestral object in Clascal (which it isn't), then all other objects would be more specific examples of the class "dog." A "poodle," for instance, would be defined as a dog of a certain size and shape with curly hair, a short tail, and so on. "Toy poodle" would be a case of the class "poodle" except of a smaller size. With these types of clear interrelationships and definitions, Clascal provides a method for programs to manipulate objects explicitly. More than that, like Pascal, it provides a structured method.

In down-to-earth terms, this means that Clascal is the first language that provides independent software developers full access to Lisa Technology. It's a shame that Clascal took so long to emerge, because it is the package that promises ultimately to make Lisa as useful and diverse a system as it has the potential to be. The appearance of Clascal may overcome the bottleneck that, so far, has prevented independent developers from creating that software for the Lisa.

Objects for a New Generation

Bruce Blumberg, formerly of the Lisa division and now a member of Apple's Macintosh marketing team, says that object-oriented programming may initially be a difficult concept for programmers to adapt to. "People who have never programmed before may grasp this approach faster than programmers already trained in the procedure-oriented methods."

To make it easier for programmers to reach this mind set, Apple is providing the Toolkit for developers to start with. The Toolkit is a generic application program combined with a set of building blocks for specific types of applications.

So just what is a generic application program? It is certainly not a plain-label database available at the supermarket for a discounted price. Rather, it is the Clascal programming necessary to implement those things that Lisa applications have in common. Lisa applications have standardized ways of dealing with menus and windows and standardized ways of reacting to things the user does with the mouse. They share the same desktop filing system and trade information on the same Clipboard. The generic application contains all the code necessary to do those things. On this base, any Lisa application program can be built.

Some applications won't require all of those things to behave exactly the same. It just isn't practical for a database to act exactly like a word processor, nor is it meaningful to select a paragraph within a graphing program. Because of this, programming in Clascal with the Toolkit is a lot like sculpting. The artist starts with a block of granite and a chisel and sets out to chip away everything that is not Art.

In addition to the generic application, there are sections of code called building blocks. These are still generic in a sense, but they are more specific to certain kinds of applications. For instance, there might be one entire set of building blocks dedicated to databases and another set dedicated to word processors. While we don't want all word processors to act exactly alike, it makes a certain amount of sense to have similar functions behave similarly. Programmers can pick and choose from among the building blocks to build custom programs with minimal effort. Those things that need to be different can still be different. In addition to building blocks and the generic applications, programmers still have the entire Clascal language with which to customize their work.

develop software for Lisa or any other system.

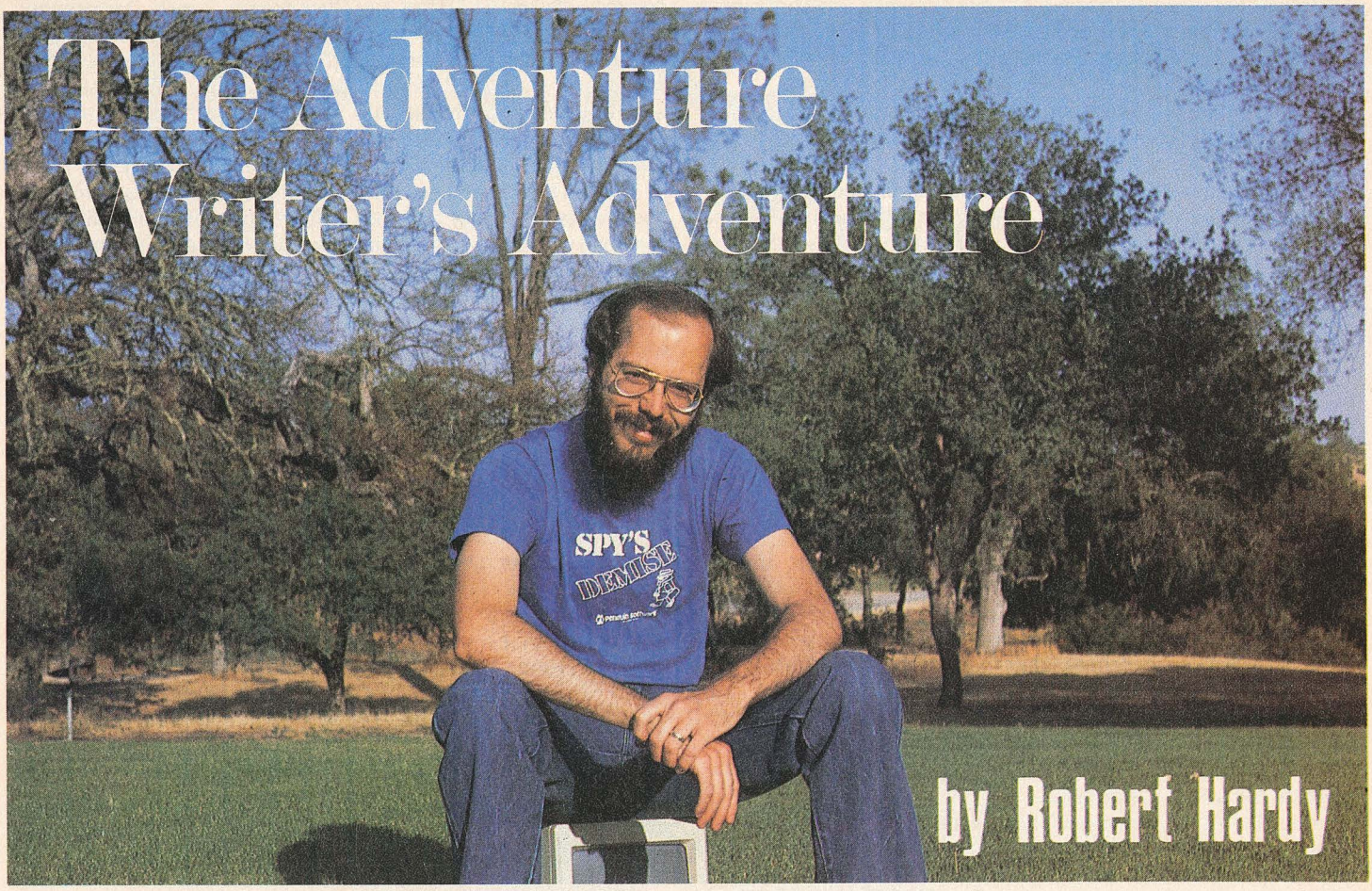
One capability Toolkit adds to Lisa's windowing system that didn't exist in the earlier applications is split windows. Programs developed with Toolkit will be able to display numerous parts of the same file within a window split horizontally, vertically, or both ways many times. Each of the window "panes" created by splitting a window will have scrolling panels that are independent of the other panes.

The fact that Clascal and the Toolkit will be available only to developers does not mean that they will be completely unavailable to someone who wants to break into the industry. It does seem to mean that Apple wants to keep an eye on what is being developed, at least at first. Burt Cummings, a veteran of the Lisa division and now the program manager of Apple university sales, says that people who want to become Lisa developers should contact Apple. They aren't necessarily looking for a proven track record in software publishing. They are looking for people or companies with good, well-conceived ideas and some evidence that they have the wherewithal to pull them off. A licensing fee is required, but not until a product is ready for distribution.

While Apple's development support group is making Toolkit available to independent developers, they aren't pushing it. They aren't encouraging developers to write software exclusively for Lisa. In fact, they aren't even giving technical support to those who insist on using Toolkit and Clascal. Lisa product marketing manager Randy Battat wants to present a unified thirty-two-bit product line. To this end, Apple is favoring Macintosh product development over Lisa product development and, on the Lisa, favoring MacWorks over the Lisa's Office System.

The future of Clascal and the Toolkit is admittedly hazy. Battat says that both are remarkable achievements in software engineering. The last word, for now, is that Apple is investigating the possibility of implementing them on the Macintosh. Until then, developers wanting to work only with Lisa can use the tools Apple has provided—but they're doing so on their own.

The Adventure Writer's Adventure



by Robert Hardy

I was working the Penguin Software booth at the winter Applefest when Guy Kawasaki, sporting an Apple badge with the title "Software Evangelist," approached us to join Apple's Macintosh prerelease software development team. Mark Pelczarski, president of Penguin, asked if I would like to do *Transylvania* as our first Mac product. *Transylvania* is a graphic adventure game originally written by Antonio Antiochia for the Apple II.

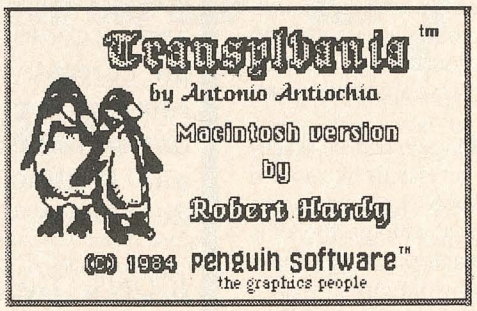
At that time the sum total of my knowledge of the Macintosh was limited to the prevailing rumors—that it used a 68000 processor, 128K of memory, a 3 1/2-inch Sony disk drive, and a mouse. Since I had just spent the last year programming the 6502 processor, I figured that it would provide a nice change of scenery, so I said, "Yes!"

I agreed to meet Guy in Cupertino the following weekend for my first introduction to the Mac. In the meantime, I studied everything I could lay my hands on about 68000 assembly language programming. Having spent a long time working with the three eight-bit registers of the 6502, I found the sixteen thirty-two bit registers of the 68000 to be pure nirvana.

Boy Meets Mac

When Guy showed me the Mac in Cupertino the following week I was initially disappointed by its diminutive size. I drove four hours to see a toaster with a built-in TV screen? But as Guy started pulling down menus, pointing, dragging, clicking, and double-clicking, the Mac began to grow until it completely filled my mind with its enormous possibilities and potentials. By the time I saw *MacPaint* I knew I was MacHooked!

Guy told me he could send everything I needed to create Mac products in four to six weeks. I knew that I'd be climbing the walls, waiting each day for the UPS truck.



Programmer Resorts to Manual Labor

Finally the Mac arrived. It was accompanied by a box that contained the development software and the documentation. The documentation was in the form of three stacks of manuals that contained everything I needed to know and was not afraid to ask. The manuals, collectively called *Inside Macintosh*, held information concerning everything from setting up the Lisa to creating programs on the Lisa and transferring them to the Mac.

Unfortunately, these manuals were simply prerelease photocopies, so my first afternoon was spent punching holes and organizing sheets of paper into more than twenty color-coded binders. (Now, when I need to refer to the QuickDraw manual, I just reach for the yellow one and so on. It's a real timesaver.) Most of the documentation contains descriptions of the Toolbox, the aptly named set of routines that reside permanently in the 64K of ROM in the Mac. *Inside Macintosh* is absolutely essential to the development of any program for the Mac.

QuickDraw MacDraw Meets the Werewolf

The making of Macintosh *Transylvania* started with the conversion of the original version's pictures. The pictures were created on the Apple II with Penguin's *Graphics Magician*, a utility that allows you to create drawings on the screen and save them to disk in a special compact format.

Graphics Magician does not save the bit map of the picture but rather the individual steps that were used to create it. The heart of the picture editor is called Picdraw, not to be confused with QuickDraw (the graphic routines in the Macintosh Toolbox). Picdraw was designed by Mark Pelczarski to operate very compactly and efficiently on the much-less-powerful 6502-based Apple II; it predates both the Mac and the Lisa. Picdraw basically does three things: It draws lines, plots brushes, and fills areas with patterns. It must do these things

Robert Hardy is a free-lance programmer who wrote the Macintosh version of Transylvania for Penguin Software.

very quickly, as some of the pictures in *Transylvania* contain more than five hundred steps.

To write Mac *Transylvania*, I had to convert these routines from 6502 to 68000 assembly language and modify them to run on the Mac. QuickDraw contains some very fast fill routines, so you may wonder why I didn't take advantage of them. The answer is, although there are incredibly fast routines to fill rectangles, circles, and even arbitrary areas called regions in QuickDraw, all of them work only on predefined areas. That is, QuickDraw must know ahead of time exactly where the boundaries of the area to be filled are. Picdraw needs to be able to fill an area whose outline isn't known until fill time, and it has to be as fast as possible. Using QuickDraw fills was out; I had to resort to good old assembly language.

Once the picture editor was usable it was time to transfer the original Apple II files over to the Lisa. This was a simple matter of making a few changes to the picture listing utility provided by the *Graphics Magician*. This utility allows you to print out the commands in a picture file so that you can see exactly what steps are taken to create any given picture. I modified it to convert each individual step into 68000 assembler–ready format and transfer the results to the Lisa.

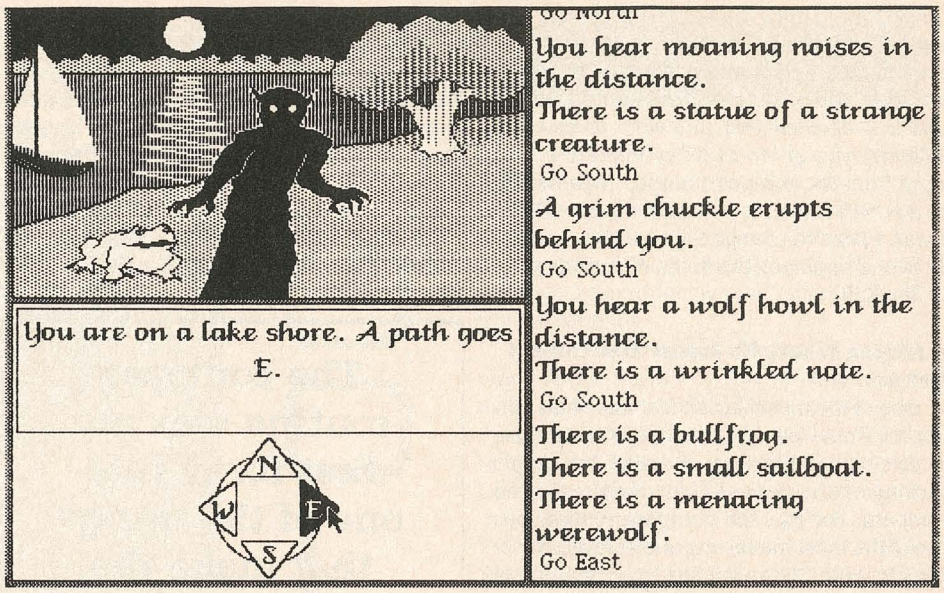
Transferring files for the eighty-plus pictures in *Transylvania* took a day and a half of pure tedium. It would have taken a lot less time if the Lisa software had been able to talk to my Apple II faster than 1200 baud. Even at that speed I had to retransmit an occasional file because of missing characters. Alas, one of the dangers of being among the first to work with a new computer is getting cut by the sharp edge of its technology. The new Lisa 2 Workshop environment has eliminated this problem.

After the files were transferred to the Lisa they were assembled individually and collected into one large resource file. A Macintosh resource file can contain many different types of data including pictures, text, object code, or whatever you decide to put in them. Resources can be retrieved from such a file very easily by making calls to a set of Toolbox routines called the Resource Manager.

In the Apple II version of *Transylvania* there are over eighty files containing one picture definition each. On the Mac there is one file containing over eighty picture resources. When I need to draw a picture, all I have to do is make one call to the Resource Manager to get it. I make one more call to Picdraw and there it is on the screen. Who says there's no Santa Claus?

Any Color You Want as Long as It's...

This file and the Mac *Graphics Magician* picture editor were sent back to Penguin in Illinois where Holly Thomason, coauthor of the *Coveted Mirror*, edited the pictures into their final form. This was necessary because the original pictures were done in color. The numbers used to represent colors internally needed to be changed to the numbers used to represent the black-and-white patterns of the Macintosh. It took a graphic artist like Holly to select the most visually pleasing patterns for each of *Transylvania*'s locations. I was really pleased when I saw the pictures.



While Holly was busy editing the pictures, I began the task of converting the main logic of *Transylvania* from Applesoft Basic to 68000 assembly language. It took me four ten-hour days to type it in. I would read a line of Basic, convert it in my head to a routine in 68000 assembly language, and then type the result into the Lisa using the mouse editor. This may sound impressive, but the truth of the matter is that the majority of the main logic was so similar that once I struggled past the first few pages my eyes and fingers went into autopilot and my mind went to sleep. Actually, I'm lying—it was very hard work for me; so go ahead and be impressed.

Will the People Please Rise

At this time, the afternoon mail brought a gift in the form of a summons. No, I wasn't wanted for conspiracy to commit computer crime or even for jaywalking. I was wanted for jury duty. Just what every free-lance programmer needs—something to take him away from his contracted work to sit around for who knows how long with eleven strangers to decide what to do to another stranger. On the drive to town I tried to think of an excuse that would get me off the hook. Maybe I would say, "Your Honor, I am the duly appointed Terran ambassador to the Bogan Empire and if I am not at the spaceport in five Earth-ticks, heads will roll!" Maybe I wouldn't.

As those of you who have been called to serve as a potential juror know, there is a tremendous amount of waiting involved. I used this time to edit the program I had just typed in. I looked for (and found) lots of typos, syntax errors, and just plain dumb mistakes. A nice, quiet, well-lit place is wonderful for this type of work, and I got this phase of the project done in a much shorter time than if I had done it at home. At the end of each day I was told to go home and call back to see if I would be needed the next day. This went on for a week and I never did get on a jury, but I didn't mind. I was well paid for my efforts—seven dollars a day.

The Parser Gets Munged

When you play *Transylvania*, you communicate with the computer by typing in simple two-word commands. The routine that deciphers these two words is called the *parser*. The word *parse* means to analyze grammatically, and that is exactly what the parser does. It looks up the commands in a table containing close to three hundred words. When it finds a match, it returns the numerical position of the word in the table. This number is then used by the main logic to determine the appropriate response or action. This is actually a very rudimentary form of artificial intelligence, which is why adventure games can be entertaining for an extremely long time.

I used one of my favorite routines in the Toolbox to form the main body of the parser. It is a single call by the name of Munger, which can do a plethora of wonderful things with data. If you tell Munger where your word table is in memory and give it a word, it will tell you where it exists in the table or it will tell you it couldn't find it. Of course, that's only a small part of what Munger can do, but, needless to say, it made short work of writing the parser.

And Don't Forget the Windows

After the parser was written and tested, I decided to tackle the text output. One of the three windows displayed during the game is devoted to both the input of the player's commands and the output of the game's responses. I wanted to be able to type in the commands at the bottom of the window and have the whole window scroll up as the program printed its responses. There is a set of routines called TextEdit that allowed me to accomplish this easily.

TextEdit makes it possible to do absolutely amazing things with text. TextEdit calls

QuickDraw to draw the text on the screen. The text output routine automatically adjusts to the extra screen height and width that becomes available when you run *Transylvania* on the Lisa 2. I decided that, if I were to use two different type styles to differentiate the user input from the computer output, there would be no confusion as to who said what to whom. It was a snap to change the type styles on the fly by manipulating the fields of records used by TextEdit.

A Region Where No Mouse Has Gone Before

One of the most Macish features of my version of *Transylvania* is the use of the mouse to select the direction of travel by simply clicking on the desired point of an on-screen compass. For me, the compass routine was one of the most interesting and enjoyable sections to write. It was my first foray into one of the things that make the Mac insanely great—regions.

A region is a QuickDraw data structure that is a collection of connected lines and framed shapes. In addition, a region can contain other regions. What you end up with is quite simply a mathematical representation of the outside boundary of an arbitrary area. You can fill regions, shrink or expand them, clear them, draw their outline, calculate their intersection with other regions, detect whether a given point (such as the mouse pointer) is in them, and do many, many other fabulous things.

All I had to do to create the frame of the compass was to open a region and call other QuickDraw routines such as LineTo and FrameOval. While a region is open, no actual drawing takes place on the screen. Instead, the end points of all the lines are stored into the region data structure. Then, whenever the compass needs to be drawn to the screen, it's a simple matter of making a single call to FrameRegion and four consecutive calls to DrawChar to letter the four direction points.

The compass routine was my first foray into one of the things that make the Mac insanely great—regions.

During game play, the routine PtInRgn is called to determine when the mouse has been clicked in a compass point, and the routine InvertRgn is used to show which point you have selected. There is a set of Toolbox routines called the Control Manager that can be used to design and manipulate compasses, switches, volume controls, or anything else your heart desires. You can even store them in resource files and use them easily from different programs.

Zen and the Art of Computer Programming

The Macintosh is really a different kind of computer to program for. It requires a different type of programming philosophy. It was necessary for me to unscrew my head, empty out a few antiquated thoughts, and screw it back on before I felt at home with my mouse-sporting friend. In most other microcomputers, the program must spend most of its time locked in a loop where it knows exactly what kind of thing is going to happen next. Let's call this thing it waits for an *event*. The programmers who write the code for these other micros only need to deal with one kind of event—the keypress. All they need to have the computer do is collect enough of these events, see what it all means, and call the appropriate subroutine to do the desired task.

On the Mac, there are many types of events that can occur at any time. Mouse-down events occur when you click the mouse button. Activate events are generated when windows are opened or change in their front-to-back ordering. Update events request you to redraw the contents of a window. There is even the lowly keydown event with its reminder that, even with mice available, people still have fingers to poke things with.

All of these events are recorded into an event queue. It isn't strictly a matter of first come, first served when it's time to see what's next in line. Some events have a higher priority than others and can be found taking cuts in front of the other events in the queue.

A Macintosh program consists of a main loop that must be able to handle the events given to it by the Event Manager, make a call to the appropriate subroutine, and loop around and do it again. This ability to take things as they come is what allows you to open several windows on the Mac and do completely different operations in each of them.

One of my biggest difficulties was becoming familiar with this type of thinking. I'll bet dollars to doughnuts that going back to programming non-Lisa machines—perish the thought—would be even harder still!

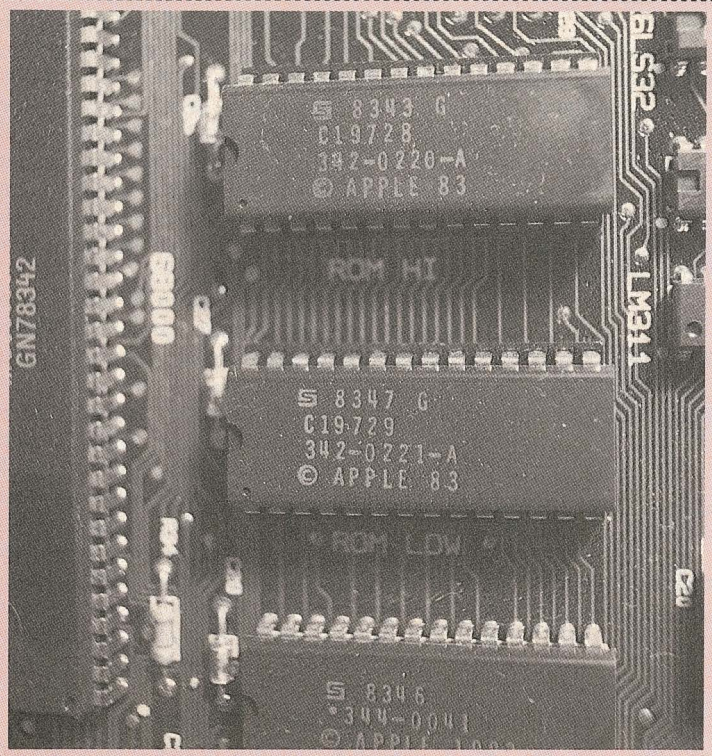
The main loop of *Transylvania* is the only portion that is written in Pascal. Pascal is ideally suited for this kind of programming because of its tendency to force you to write no-nonsense code. As I had never written in Pascal, there was an incredible amount of trial and error required to get the darn thing to work right. If it weren't for Cary Clark and the others in the Macintosh Technical Support Group, *Transylvania* would most likely not be available yet. On the other hand, my phone bill wouldn't have to be delivered by truck. Thanks, guys.

Light the Fuse and Stand Back

Once all the subroutines had been written and the main loop was more or less in shape, it was time to put the pieces together and see

TOOLBOX

By Robert Hardy



Inside the Macintosh sits 64K of read-only memory (ROM) known as the User Interface Toolbox. This Toolbox contains a library of more than four hundred subroutines that are easily available to the programmer. These painstakingly well-thought-out routines take an incredible burden off of Macintosh programmers by allowing them to concentrate on the application itself, rather than the interface to the user.

The following descriptions of the more important Toolbox units are by no means a complete account of what is contained within the Toolbox. They should, however, give you a taste of the great things already built into each Mac that are just waiting to be set free. For more detailed descriptions of these routines, refer to the User Interface Toolbox manual.

Control Manager. The Control Manager is used to create, display, and manipulate the unique, interactive controls that make the Mac so simple to use. The standard controls can take the form of scroll bars, check boxes, and buttons. Nonstandard controls such as compasses can be custom-designed by the application programmer. The Control Manager contains approximately twenty-six subroutines.

Desk Manager. The Desk Manager handles the desk accessories such as the clock. From a programming point of view, the Desk Manager is almost transparent. It contains approximately seven subroutines.

Dialog Manager. The Dialog Manager is responsible for the creation and handling of dialog and alert boxes. Dialog and alert boxes are used to facilitate the interaction between the user and the application. It contains approximately twenty-three subroutines.

Event Manager. The Event Manager is responsible for keeping in touch with the outside world. It keeps track of such things as the mouse actions, keypresses, and the insertion of disks. Internally, it generates events used in the management of windows and also allows you to read the system clock. It contains approximately eleven subroutines.

Font Manager. The Font Manager is used to select the fonts used by QuickDraw. It can also return detailed information about a desired font to the application. It contains approximately five subroutines.

Menu Manager. The Menu Manager is responsible for the creation and manipulation of the menu bar. It can also handle custom-defined menus such as pull-down menus containing patterns. It has approximately thirty-three subroutines.

QuickDraw. QuickDraw is a set of extremely fast screen-drawing routines that are utilized by almost every component of the Toolbox. Generally recognized as the heart of the Lisa and the Macintosh, these routines are responsible for absolutely everything you see on the screen. There are very few graphic operations that can't be accomplished using QuickDraw. QuickDraw contains approximately one hundred thirty-five subroutines.

Resource Manager. Resources are the many different types of data structures that make up an application. They include such things as text strings, pictures, menu definitions, and even the application's program code. The Resource Manager is called, as needed, by the application to gain access to the data structures stored in resource files. It contains approximately thirty-four subroutines.

Scrap Manager. The Scrap Manager performs all of the functions of the Clipboard. It handles information exchange within an application, between applications, and between applications and desk accessories: any action that involves cutting and pasting. It contains approximately six subroutines.

Segment Loader. The Segment Loader handles the loading and unloading of program segments of up to 32K. Each segment, except for the main one, needs to be in memory only if it is in use. An unused segment can be unloaded at any time. The space formerly occupied by the segment then becomes available to the application. The Segment Loader contains approximately six subroutines.

TextEdit. TextEdit is the unit responsible for the formatting and editing of text in an application. Toolbox units such as the Dialog Manager and the Menu Manager use TextEdit for the insertion, deletion, and presentation of text. It contains approximately twenty-one subroutines.

Toolbox Utilities. The Toolbox Utilities are useful routines that allow you to do such things as fixed-point arithmetic, byte and bit manipulation, and string manipulation. They also allow you to load and plot icons, load pictures and patterns, and hide the cursor if it's inside a given rectangle on the screen. This section contains approximately twenty-four subroutines.

Window Manager. The Window Manager is used to create, display, and manipulate windows. All operations such as dragging, sizing, opening, closing, scrolling, and disposing of windows are performed by these routines. The Window Manager is also responsible for the front-to-back ordering of application and system windows. This manager calls on QuickDraw to display the windows and their contents. The Window Manager contains approximately forty-three subroutines.

Actually, we don't call them bugs anymore. The more polite term is "undocumented features."

if *Transylvania* would fly. When you've been working several months on disjointed sections of a program, you can lose sight of the overall picture of what you're trying to accomplish. It was real exciting to plug it all in and actually play *Transylvania* on the Mac.

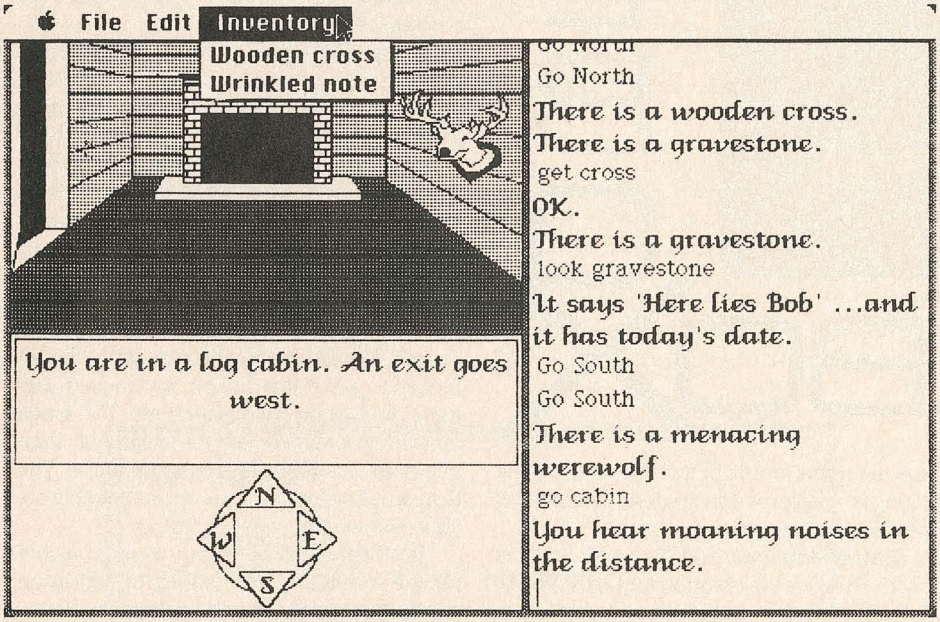
As usual, there were a lot of bugs and irregularities that didn't manifest themselves until the last minute. Actually, we don't call them bugs anymore. The more polite term is "undocumented features." With the new terminology, programmers don't have to take the blame when the program crashes right before you are about to get the highest score ever. Or when it crashes just as you finished entering the last value in a 1,024 by 1,024 spreadsheet.

Anyway, it took a large hammer and a lot of frustrating hours to finally squash the undocumented features that had wormed their way into my code. After I was sure I had found all the bugs, I sent a copy to Penguin, where Dave Albert found a few more with a high-level testing technique he has developed called "Kitty on the keyboard."

Well, Maybe a Little like 1984

I tried as hard as I could to follow the guidelines set forth by what is referred to as the Macintosh User Interface. These guidelines are Apple's justified attempts to make Mac products compatible with each other. The guidelines are one of the reasons that *MacPaint* drawings can be transferred to *MacWrite* documents. They also guide the programmer in the proper way to handle interaction between the user and the computer. If a programmer follows these guidelines, the end result will still be a product that is uniquely characteristic of that programmer. It will also be a snap to operate because the user won't be forced to learn a new set of skills for each new program.

There is only one place in the game where I failed to abide by these guidelines. Where the program asks you to sign in and enter your next of kin, I am guilty of using what is called a *hard mode*. A hard mode is where you are forced to do one thing. In my case, I require the players to enter the names in text. While the program is in this hard mode, the user can't pull down menus, edit the input field, or do anything but enter characters with the keyboard. Someone had suggested that I use a dialog box instead. I felt that this would have disrupted the overall flow of the program, so I left it the way it was—so there!



Does That Banner Yet Wave

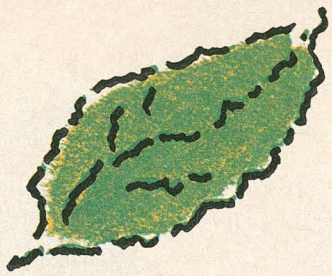
At last it was time to put in all of the finishing touches. Usually the last thing I do in a game is design the banner page. I always save it until last because I hate doing it. You know you have no artistic talent when you can't even draw a straight line using *MacPaint*. Brian Carter down at my local computer store used *MacPaint* to draw the wonderful penguins depicted on the banner page of *Transylvania*. Once he finished the drawings I transferred them via the Scrapbook and pasted them into the resource file of my program.

I'm telling you, these resources are ingenious. When I need to display the title page, all I have to do is make a call to the Resource Manager to get it and then make a call to a QuickDraw routine called DrawPicture, which actually draws it to the screen. That sounds real rough, huh?

Lead On, MacDuff

I'm currently working on my next Mac project: a strategy board game by John Besnard called *Pensate*. It's a thinking game that is superficially similar to chess, but with a personality all its own. I'm looking forward to incorporating things into it that I've learned while doing *Transylvania* and, more important, incorporating things that I have yet to learn.

I've always felt that the mind dies when the learning stops. The real joy of working with computers is that the learning never ends. *Transylvania* only scratches the surface of what can be done with a Macintosh. Working with it night and day has taught me that it's really light years ahead of other computers. It will be quite some time before the full potential of the Mac will be realized.

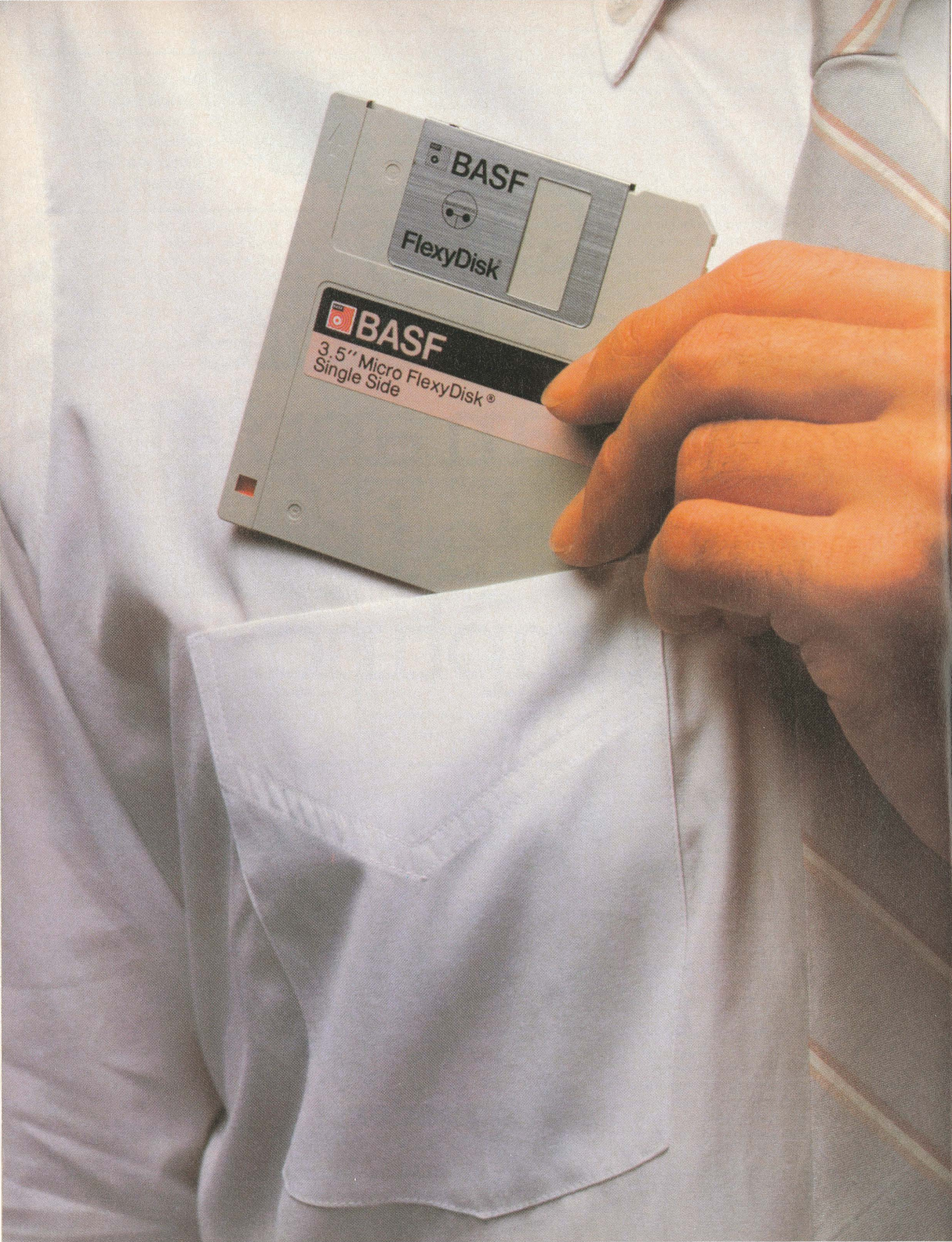


Only once before was an apple made this tempting.

Habadex for Macintosh.™ The ultimate desktop management software. It dials your phone. Sorts hundreds of records. Prints flexible reports. Produces form letters and mailing labels. And it's now running at Apple dealers everywhere.

hsi habadex™

Making Apples more tempting than ever.


BASF
FlexyDisk
BASF
3.5" Micro FlexyDisk ®
Single Side

Introducing a pocketful of miracles.

It's the new BASF Qualimetric™ 3.5″ Micro FlexyDisk.® Its name is a lot bigger than its size, and a lot smaller than its capacity – one astounding megabyte. Our new Micro fits into more than shirt pockets. It fits into several of the hottest new small portable personal computers – like Apple™ Computer's new Macintosh™ and Lisa™ 2 and Hewlett-Packard's HP-150 Touchscreen PC.

The Qualimetric part is what makes our Micro FlexyDisk different from the others that will inevitably follow. It indicates a new standard in design, production, inspection, and testing. This standard enables us to warranty the Micro for a lifetime.* No more, no less. In addition, we certify that each bit of every byte on or between every track on each Micro FlexyDisk is 100% error-free. That's because we know, while each Micro FlexyDisk may only cost a couple of bucks, the hours and brain cells you put into it are priceless.

*Contact BASF for a copy of our warranty.

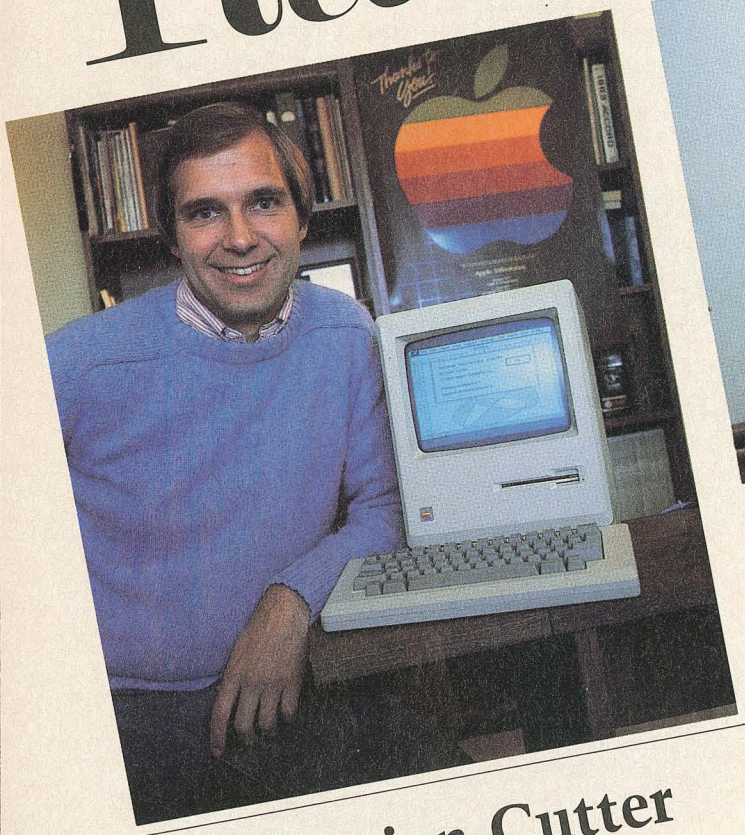
For those occasions when you need more than a pocketful of miracles, we provide a handy and virtually earthquake-proof 5-pack. It's sized to slip easily into your briefcase, handbag, or backpack.

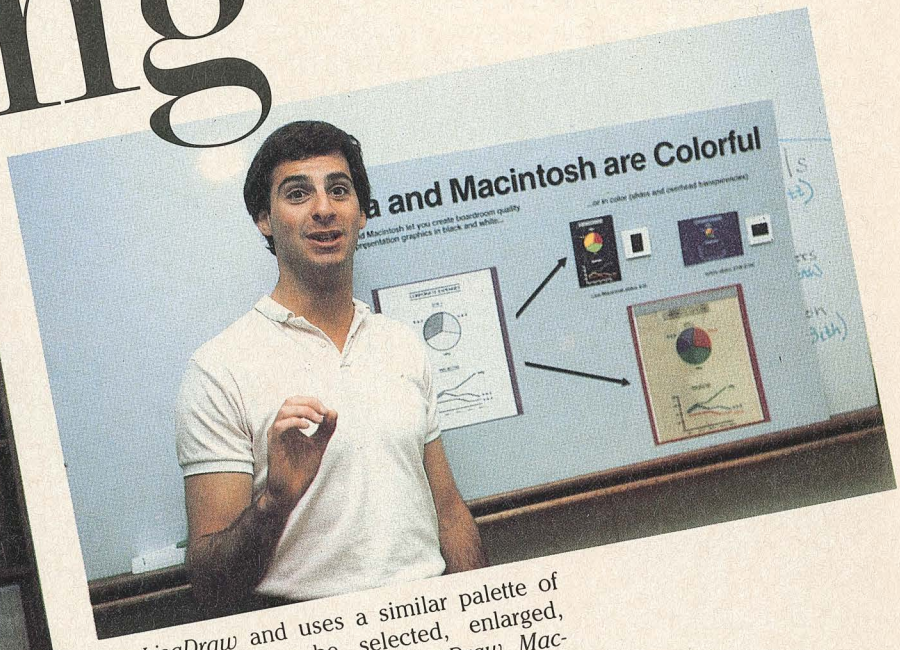
For a plethora of technical details about our new Micro FlexyDisk and even more important information on where to buy it, call toll-free. 1-800-343-4600.

Celebrating 50 years of magnetic recording

BASF

Racing





How Apple Gets Its Software Out the Door

By Brian Cutter

Have you ever wondered how a piece of software gets from idea to finished product? Who gets product ideas? Who develops them? Apple has a rather, shall we say, unique product development cycle. Since *MacDraw*, Apple's new graphics program, is nearing the end of its development cycle, let's use it as an example to illustrate how Apple develops and markets products.

MacDraw is a powerful graphics editor that allows you to compose all kinds of drawings, such as diagrams, flow charts, technical drawings, maps, blueprints, and organizational charts, from a palette of circles, boxes, polygons, lines, and text. *MacDraw* is based on *LisaDraw* and uses a similar palette of shapes that can be selected, enlarged, reshaped, and moved. Like *LisaDraw*, *MacDraw* is object-oriented; this means that a composition is represented as a collection of individual objects (shapes, lines, and text) that may be edited in groups or separately.

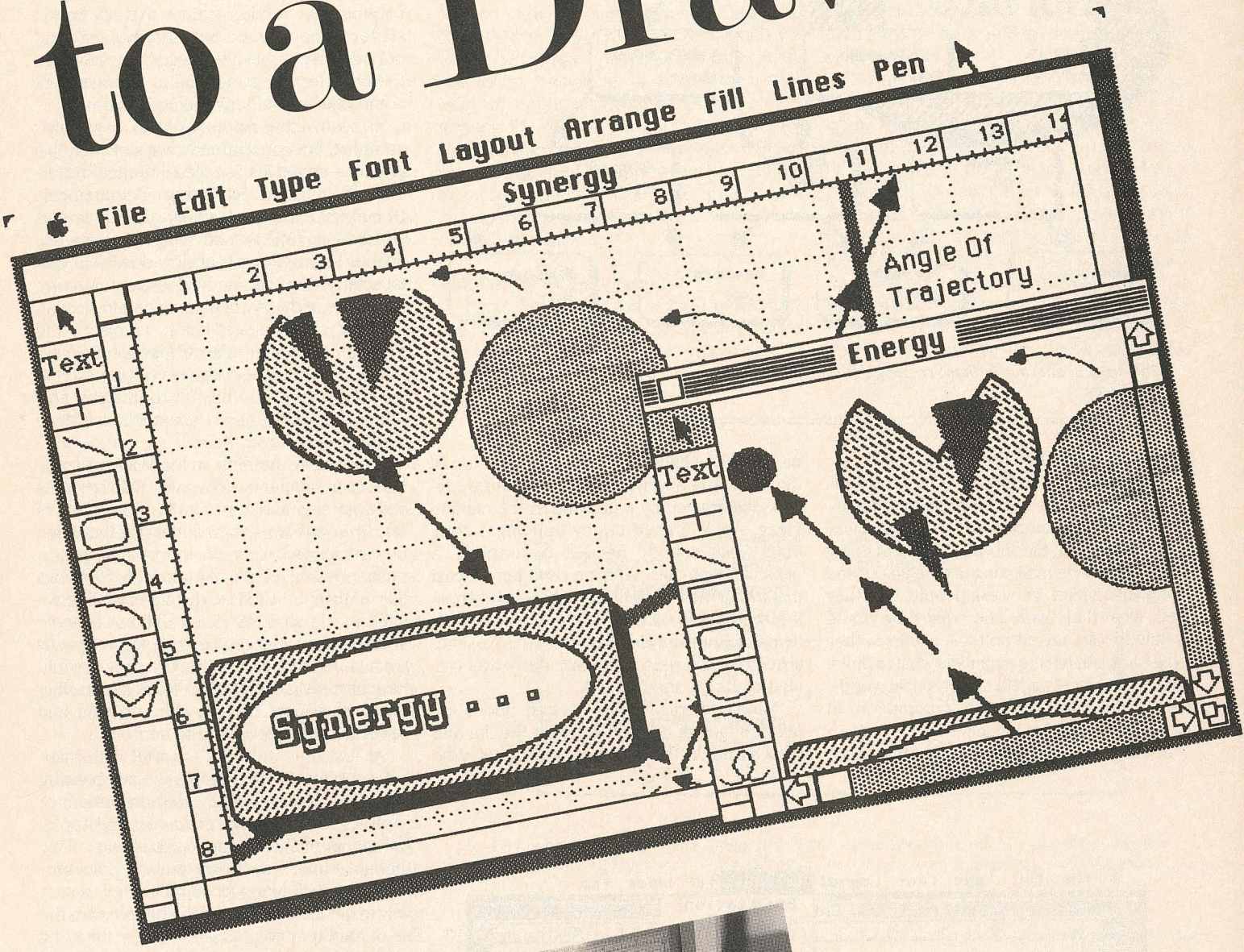
The story of *MacDraw*'s development actually begins more than three years ago with the development of *LisaDraw*, which is very similar to, and in fact a forerunner of, *MacDraw*. We can explore Apple's product development process by taking a look at the development of both programs: *MacDraw* and *LisaDraw*. Let's start with the beginning of the thirty-two-bit family, the Lisa.

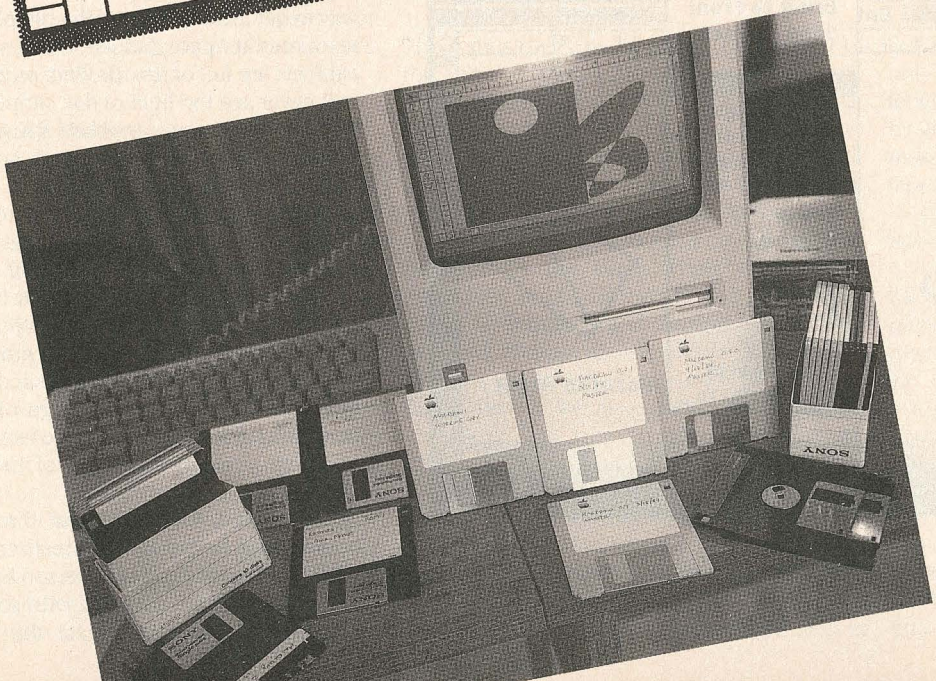
When you want to talk about the marketing of Lisa products, a good person to see is Randy Battat. As product manager for Lisa and Lisa products, Battat is in charge of setting Lisa's wholesale price, monitoring production levels, and, of course, overseeing the development of Lisa products.

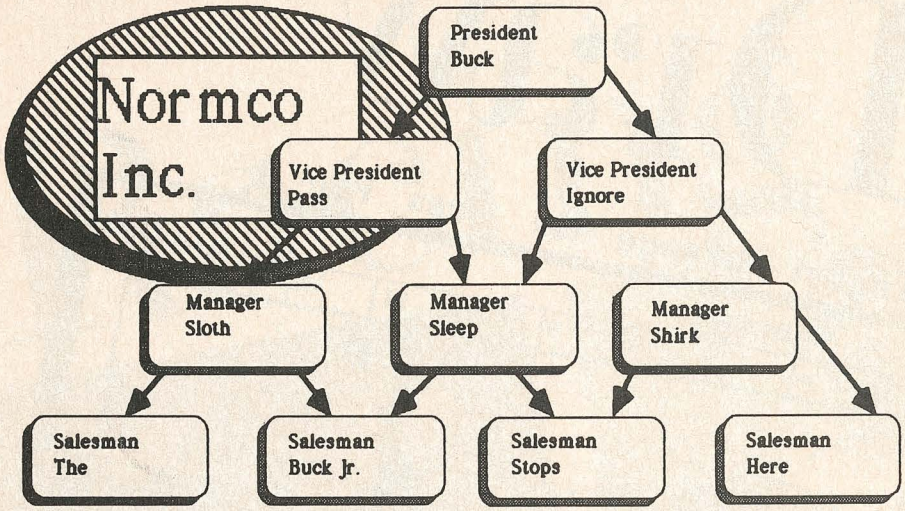
Brian Cutter is a technical writer at Dysan Corporation in the company's Software Publishing Division, as well as a free-lance writer and photographer. His brother is Mark Cutter.

Mark Cutter, above left, created LisaDraw, the powerful graphics editor upon which MacDraw is based. Lisa product manager Randy Battat, above right, believes that Apple's unorthodox marketing strategy encourages the development of uniquely creative products.

to a Draw


File Edit Type Font Layout Arrange Fill Lines Pen
Synergy
Angle Of Trajectory
Energy
Text
Text
Synergy . . .





The original directive to those creating the Draw software was to make a simple Org Chart Generator.

Deviation from the Norm

Battat begins the tale of Apple's unique method of product development by explaining the "normal," as opposed to Apple's, way of doing things. In the "normal" mode of doing things, innovative marketing people go out and survey the market. They ask people what they need, what they want, and what they would be able to use. Based on these surveys, they come back and tell the engineers what to build. Then, in that most perfect of theoretical worlds, the software engineers write programs to fit the niche the market has defined.

But we don't need Battat to tell us that Apple has a different style. Until the introduction of the IIc, Apple never designed a machine specifically targeted by a demand in the marketplace. Apple started out by building a computer that nobody needed *or* wanted. A personal computer? What on earth for? At least that's what most of us thought. When the Apple II turned out to be only slightly bigger than a typewriter and hundreds of times more useful, a need—for a personal computer—was created by its fill, the Apple II.

Similar stories could be told about the public's lack of desire for the III, the IIe, and most assuredly the thirty-two-bit projects. Who needs mice? Some critics, most notably Peter McWilliams, the author of several microcomputer books, have done a good job of pointing out Apple's inability to perceive the needs of the marketplace and to design a machine to meet those specific needs. But so what? Apple continues to build machines that no market surveys recommend, that nobody wants and nobody needs, but that everybody buys. So there's got to be something to this maverick company's unorthodox marketing strategy.

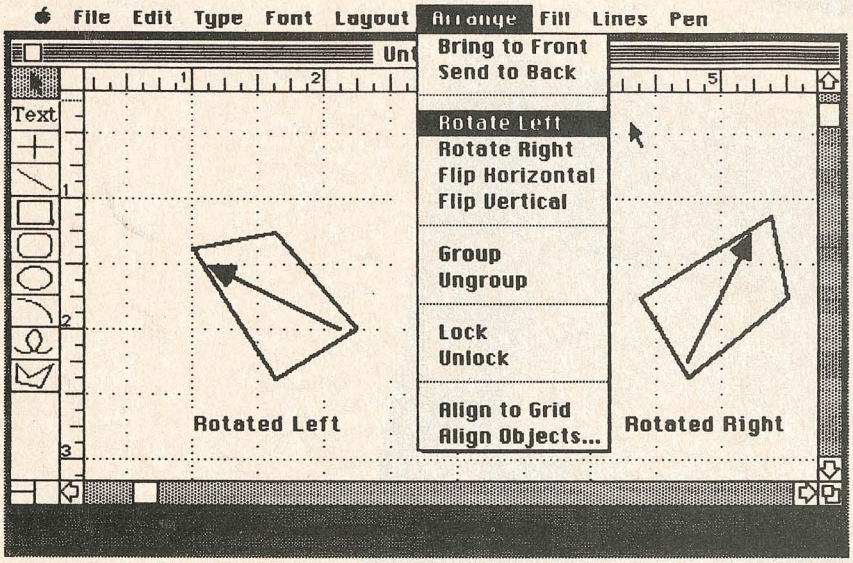
It seems that Apple used to have fairly complex, but conventional, ways of starting a project—strategies like New Product Proposals, Preliminary Marketing Requirements Documents (PMRD), Marketing Requirements Documents (MRD), and External Reference Specifications. Sounds boring, doesn't it? Battat says boring is an understatement. Anyway, a number of these documents had to be written, filed, and evaluated before a project could even get off the ground. Consequently, says Battat, "Many proposals never got off the drawing board." It's difficult to imagine anyone ever reading these documents, let alone getting excited over them.

This paper-shuffling on the *MacDraw* project was permanently done away with by Apple software engineer Mark Cutter, the author of *LisaDraw* and *MacDraw*. Cutter says that when he first started work on the project, paper-pushers hounded him day and night for things like PMRDs and External Reference Specifications. He says it's pretty difficult to write specifications before you even know exactly what you're trying to build. The papers could have been written, but Cutter knew it was either write the papers or write the code; he told Apple he didn't have time to do both.

At first, this didn't sit too well with management, but when people began passing mock-ups (simple working demonstrations of ideas for software) of *LisaDraw* around Apple, the paper-pushers stopped pushing. They thought Cutter's work was fantastic. According to Randy Battat, a good mock-up is a sure way to get the attention of the people with the resources at Apple. Indeed, he says, the filing cabinets are full of new product proposals that will never see the light of day simply because the engineers never mocked them up. So if you have an idea for a program, better mock it up before trying to get funding.

Cutter wasn't originally hired to work on the thirty-two-bit projects. He was supposed to be writing a graphics editor for the Apple III. But one day he wandered out to the Lisa building and started chatting with Bill Atkinson, who did all of the low-level graphics routines for the Lisa and the Mac and played a major role in the overall design of the Lisa, contributing to the design of the user interface and to some extent the design of the hardware as well.

Cutter and Atkinson spent three or four hours talking before Atkinson realized he wasn't talking to the new graphics person for the Lisa. Apple had been negotiating with someone to write the graphics editor for the Lisa, and



MacDraw is capable of rotating objects in ninety-degree increments only. Cutter would eventually like to see arbitrary rotation of shapes.

Atkinson thought Cutter was that person. But the person Apple was negotiating with never came aboard; so, just like the stand-in who gets to go on for the indisposed lead on opening night, Cutter soon got his wish to join the Lisa project.

(To be sure, this is only an isolated example, but perhaps you're wondering how such a successful organization as Apple can afford to go about getting its software engineers in such unorthodox ways. Randy Battat puts it this way: "Old theory," he says, "has a lot to do with structure. Structure is good," he pauses, searching for *something* good to say about structure, "...because you can measure it." Structure gives the paper-pushers something to measure, as in: "Oh, the External Reference Specifications call for an 'A' here and your program gives a 'B'; 'B' certainly works better, but you failed 'cause your spec says an 'A.'")

A Method to This Madness

Battat has worked at Apple for a number of years, and he has found that the project engineers who succeed all do certain things the same way. They stick to a methodology that has evolved over the years. And those who fail are those who ignore these methods.

According to Battat, the first thing that an Apple engineer with an idea for a product should do is "demonstrate an early complete prototype that doesn't aim to do much but that does it well." He says an idea that can be viewed on the screen is much more exciting and likely to get funded than an idea, no matter how good, on paper. Cutter, coincidentally, expressed a similar belief when describing why he decided to do a mock-up rather than a lot of paperwork: "Ideas are fine, but experience is much better."

Believe it or not, the first directive to those creating the *Draw* software was to make a simple Org Chart Generator. You know, an organizational chart—vice presidents are always drawing them up for other vice presidents so everyone will know where everyone stands in a given week. Although *LisaDraw* started out that way, it has evolved tremendously. Let's look at how it all started.

Mocking Up MacDraw

A very elaborate working prototype of a graphics editor had been built by Rod Perkins before Cutter came aboard. Cutter and Perkins spent a month or two meeting and discussing what the graphics editor should ultimately do, but they weren't able to come up with any satisfactory designs; nevertheless, they drafted a design from that first prototype. When Bill Atkinson saw it, he was decidedly unimpressed. So he went out himself and in a few days mocked up a completely different prototype.

Atkinson's prototype contained the basic ideas out of which *LisaDraw* and eventually *MacDraw* would evolve. It drew an object from a prototype palette and had handles one could use to stretch it. It was a very simple working version with only a few shapes and options. Once Atkinson had mocked it up, he went back to finish QuickDraw, the graphics primitives in the ROM, and Rod Perkins went on to write *LisaCalc*.

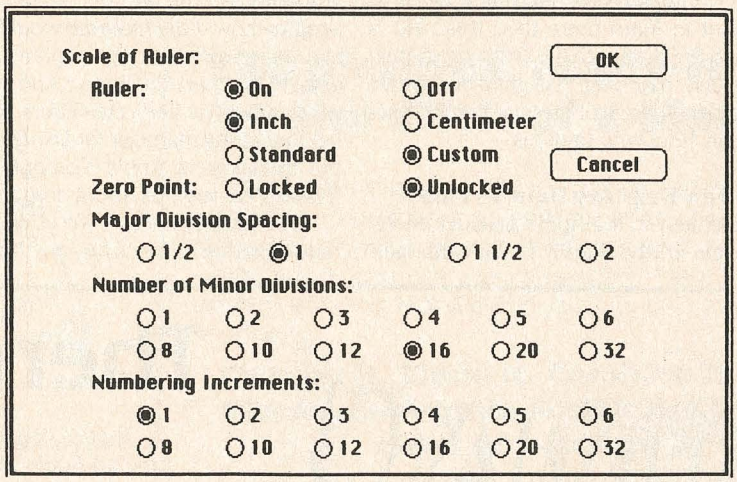
Well, the next few iterations of *LisaDraw* followed precisely Battat's first unwritten rule: "Get an early working mock-up." Cutter proceeded to get up several early working versions of *LisaDraw*. In fact, he got the next and successive versions up so fast that the usually overly optimistic marketing people had to keep revising the release date of the product *forward* in time. Originally scheduled for release six months after Lisa, *LisaDraw* was released at the same time as Lisa, despite the fact that it was started a year later than some of the other Lisa software.

The Power of Suggestion

Battat insists that his second observation about how software is most efficiently developed at Apple is just as important as the first. There must be "a willingness, on the part of the software people, to let people play with the early versions of a product and a willingness [on the part of the authors] to listen to what they have to say." In other words, an engineer has to make duplicates of his product, pass them around Apple, and then maintain a constant openness to revision. He believes that this is the stage where creativity is embedded within the project, as a direct result of the harmonious and sharing environment at Apple. Without mentioning any names, Battat goes on to tell about products that had failed, he felt, because their authors had an attitude of, "Oh, I don't want to show it to you; it's buggy." Many good suggestions have been the result of releasing a product in-house.

The ability to lock down an object in *LisaDraw* and *MacDraw* was requested by a frustrated early user who accidentally kept selecting the wrong object when trying to manipulate very small graphics right next to each other. That lock-down feature, as well as the movable zero-point, were suggestions from Apple's own Wayne Rosing. Rosing managed the hardware development of the Lisa and also was technical manager of all of the engineers for the Lisa project. In Cutter's view, "Wayne is one of the greatest managers and executives because he would go home and *use* the software."

Rosing was adding a shop to his home and took an early version of *LisaDraw* to use in laying out the shop. It was while using the program to design the workshop that Rosing wished some of the objects could be locked down. This was also when he felt the need for a movable zero, or reference, point. A movable zero point allows the user to show one ruler for one area or object in a drawing and then turn on a new ruler with a new reference point at another location. It works like a tape measure. When you want to measure a wall, and then measure the length of a desk against that wall, you don't stretch the tape measure out, leave it, and then try to determine the length of the desk by subtracting it from the length of the wall. You simply bring the tape measure to the edge of the desk. Well, like a tape measure, *MacDraw*'s rulers can move too.



A movable zero, or reference, point allows the user to show one ruler for one area or object in a drawing and then turn on a new ruler with a new reference point at another location.

One suggestion for *MacDraw* came from a rather unusual source. It might have come from one of you; Cutter doesn't know the name of the person who made this suggestion, but he's grateful. Cutter had volunteered to answer questions at the West Coast Computer Faire for Apple. A patron who had been using *LisaDraw* for some time suggested, "Why not show the size of the arc when RESHAPE ARC is turned on with RESHAPE POLYGON?" Cutter thought that was a great idea and went home and implemented it immediately on *MacDraw*. (So next time you go to a computer show, be sure to ask whether the product you are trying out is under development. If it is, and you can make a worthwhile suggestion,

you might just see your idea implemented.)

Cutter says he would like to implement "Undo," which is on *LisaDraw*, but he doesn't think he will have time to do that before *MacDraw* is released. He would also like to implement text rotation, but once again there might not be enough time before his deadline. These are both bells and whistles that might or might not make it in by *MacDraw*'s release, but at press time it wasn't known whether they would.

Cutter would also like to see arbitrary rotation of shapes. Right now *MacDraw* supports ROTATE RIGHT or ROTATE LEFT in ninety-degree increments only, but suppose somebody wanted to rotate just a few degrees. That capability would be fairly simple to implement for most shapes, but it would require new low-level graphics to work for ovals. It took Bill Atkinson several years to implement his low-level graphics routines. Anyone want to try to beat his time? Cutter would like to see 3-D implemented by someone else. "I could certainly do it, but I wouldn't be learning anything. I've written this graphics editor twice....I think I want to learn more than this." He is pretty much finished with the *Draw* programs; someone else has been assigned to update *LisaDraw*, and there will probably be another engineer on *MacDraw* updates.

The Last Few Bugs Are Hard To Find

The next step in the Apple product development cycle is what Randy Battat calls "gettin' it out the door." It's the step that many amateur programmers often ignore. If you've ever designed a program on your own, you may have gotten it just so bug-free and then left it. Commercial programs certainly shouldn't be released with *any* known bugs. Battat puts it mildly when he says that getting a product bug-free takes "a lot of blood, sweat, and tears and guts-crankin' late nights."

Debugging a program can be a frustrating and time-consuming task, and Cutter says that the last few bugs are always the hardest ones to find. A good example of almost not "gettin' it out the door" was the *MacPaint-MacWrite* disk. It seems that the final tweaks weren't applied to the master disk, from which thousands were to be duplicated, until just *two hours* before the actual duplicating began.

Battat believes that software engineers should set firm deadlines throughout the process of software development. People work better under a little bit of pressure. Programming is a creative process and therefore is subject to problems common to the arts: problems like the lack of inspiration. Can you imagine how dull *MacDraw* would be if Cutter had stayed with the original plan and had written an Org Chart Generator! And software arts is a competitive field. Dan Eilers, former product marketing manager for *LisaDraw* and now the treasurer of Apple Computer Inc., says "Always be nervous about competitors."

In the opinion of one West Coast Computer Faire attendee, "They'll buy anything that even boots on the Mac." This attitude of "write something quick and dirty and let's make a buck" comes from the fact that traditionally the largest sellers in software have always been the early birds, because there are so few available products at first. This creates fierce competition for companies like Apple that try to commit to quality software.

An analogy: Suppose someone was able to manufacture a plane that could fly at almost the speed of light. Now suppose it could be sold for the price of a Volkswagen. Finally, suppose there were only three airstrips in the world long enough to land such an airplane. A couple of things are going to happen. First, those airstrips are going to be pretty popular places. Second, the owners of the airstrips can charge whatever they please to people who land there; and third, you're going to have a mad rush of people going out and trying to lengthen their runways.

"Slipping the date" is what it's officially called when any of the many steps a product has to go through doesn't get finished in time. Given the above analogy, the Mac is the plane and the software is the runway the plane uses.

A lot of people out there are starting to extend the old, or build new, runways. Dan Eilers recently summed up his philosophy in the competitive software game by saying, "If I've got software, I'd say ship it." In the coming months, we'll see whether the Macintosh software development team did indeed "ship it."

File Edit Goodies Font FontSize Style

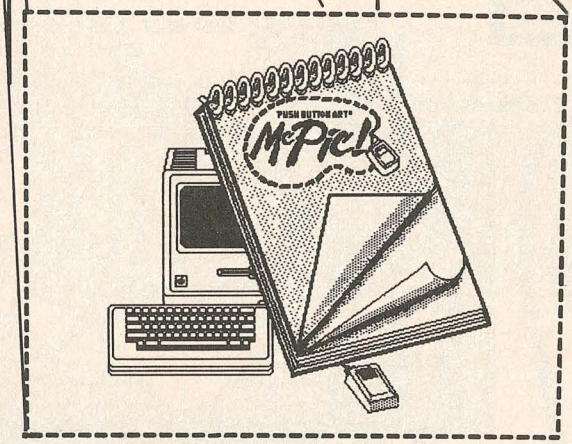
Scrapbook
Alarm Clock
Note
Calcul
Key C
Contr
Puzzl

McPIC!

"PUSH-BUTTON ART"

McPic!

For the Macintosh*

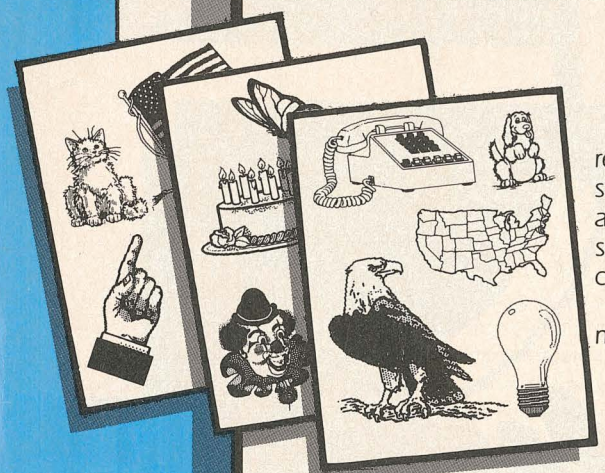


130 pictures, ready-to-use at the Click of a Mouse!

Make the most of your Mac's graphic capabilities with McPic! You don't need to have an artistic bone in your body. Here's push-button art — 130 pictures by professional artists which you can use immediately "as is". Or, with MacPaint** and MacWrite** you can take a **McPic**ture and easily . . .

Change it. Clone it. Combine it. Add Words. Enlarge it. Shrink it. Change tones or patterns.

With your imagination (and the McPic! Hint Manual) you can customize them in a thousand ways for business, home, school, recreational, club, or any other use.



Now, you can have special business letterheads, make more effective news releases, bulletins, even sales promotion items. You can create personalized stationery for every member of the family and for every occasion — design attractive party invitations — make your own greeting cards — use it to illustrate school reports — to add graphic emphasis, or humor, to your letters and other correspondence! The opportunities are endless.

Get McPic! today, and look like a talented artist in minutes. It's probably the most enjoyable and useful program available for the Mac today.

$49.95
For Macintosh with
MacWrite/MacPaint
1 Disk + Hint Manual.

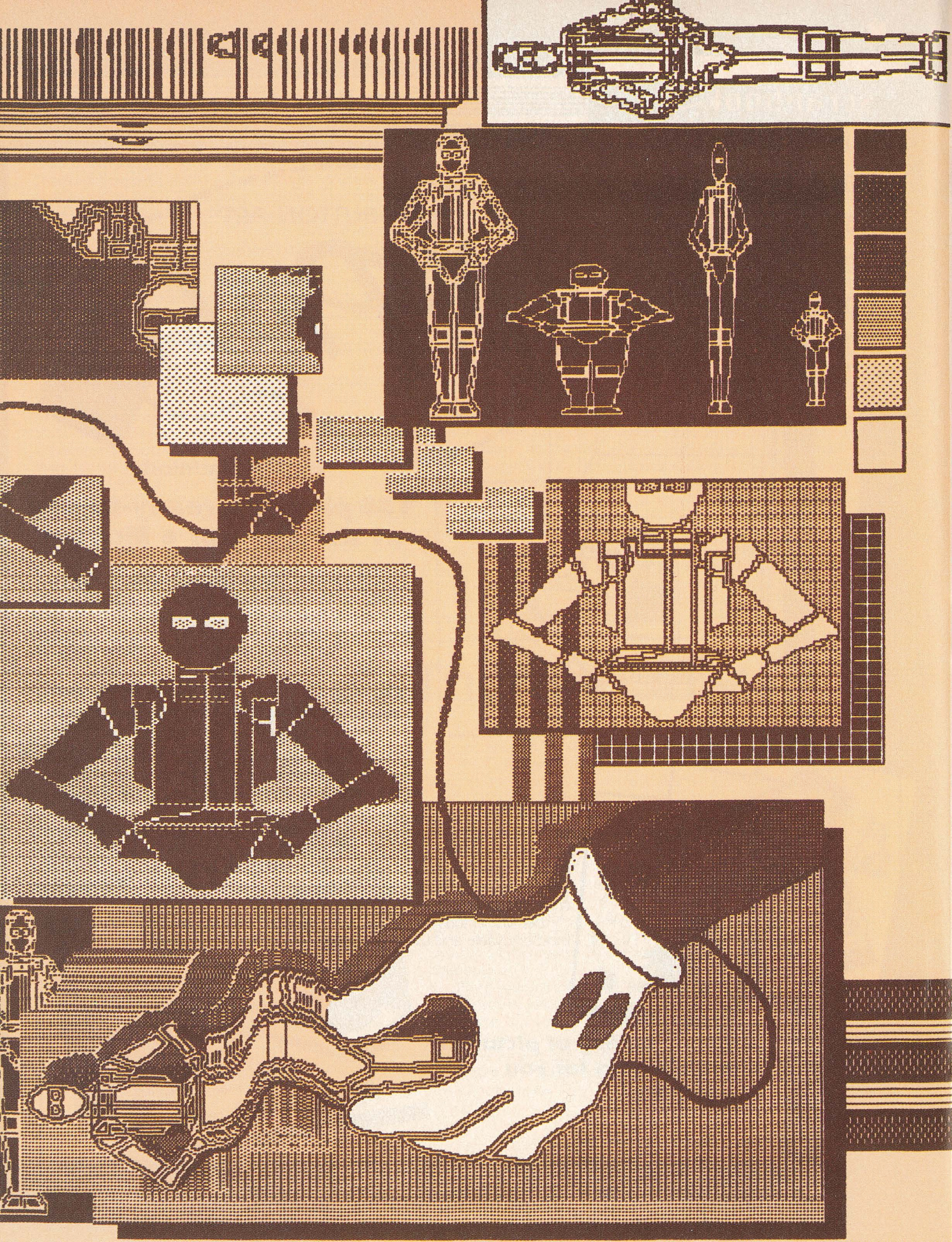
VISA/MASTERCARD/C.O.D. accepted.
Add $2.00 shipping/handling.
(Cal. residents please add 6 1/2% tax)

A terrific variety of pictures!
A million ways for you to use them!

At your dealers now, or from:

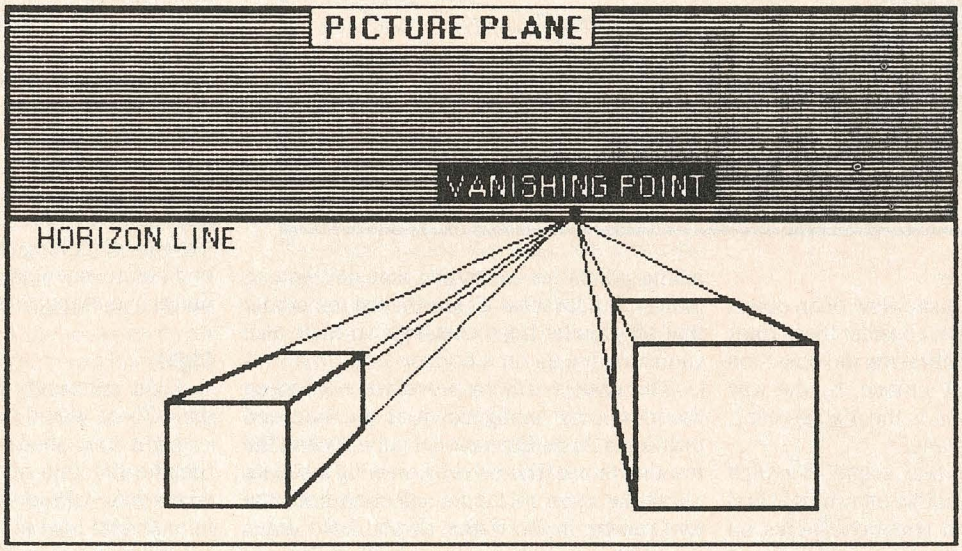
Magnum SOFTWARE
21115 Devonshire St., Ste. 337
Chatsworth, CA 91311 (818) 700-0510

*trademark licensed to Apple Computer, Inc.
**trademarks of Apple Computer, Inc.



By CRAIG AND NANCY CALSBEEK

Renaissance Mouse



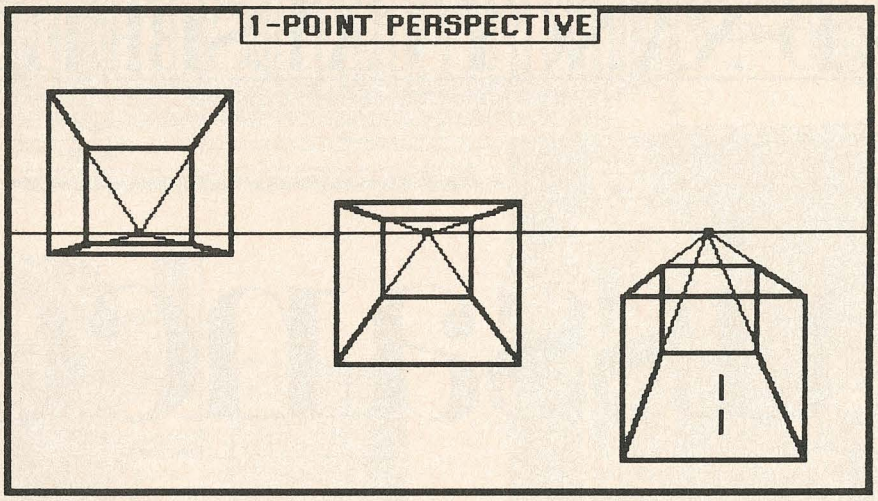
If you thought last month's installment was a little basic and you've now got a stack of printouts with nothing but circles, squares, and triangles on them, hang tight. There are reasons for learning those basic concepts. The reasons won't become apparent until you catch yourself looking at things just a little bit differently than you did before. The natural laws of light, form, and perspective are there for those who can appreciate them, and, as you do when you're learning a language, you have to trudge through the basics, memorize them, and then forget them. At that point you embrace the language and it becomes a part of your speech and your form of expression.

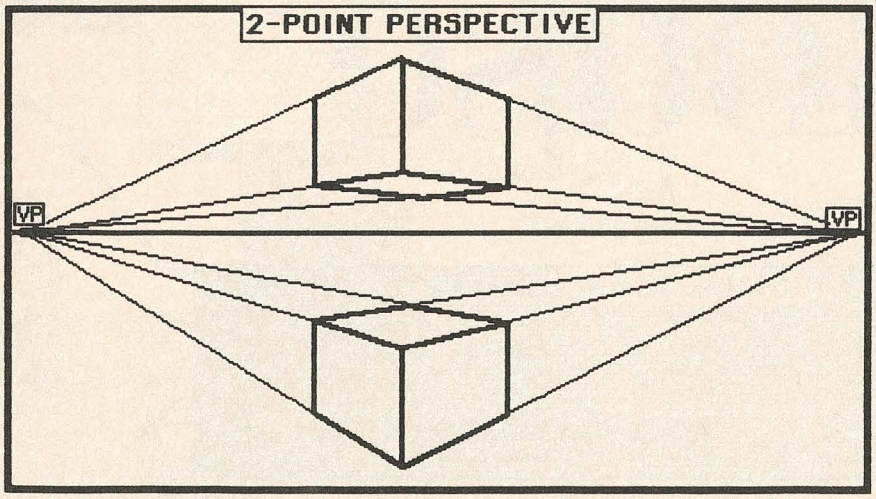
Last month we took a look at our eyes and how they work to give a person his or her own unique vision of the world. What we didn't discuss was perspective and the nature of light. This month's installment will illuminate that subject and bring you a palette of perspective tricks that you can use to give accuracy and reality to your drawings.

Perspective

Perspective is the process of representing on a flat plane the spatial relationships of objects as they appear to the eye. Simply put, as things get farther away, they look smaller. If you were to draw a straight line out from your eyes in whatever direction you happened to be looking, this line would be called your "line of sight." And the spot at which you were standing while staring off into infinity can be labeled the "station point."

Your vision is restricted to a total range of approximately sixty degrees from the bridge of your nose. It's as if you had this imaginary cone strapped to your face, with the apex of the cone starting right smack dab between your baby blues. Anything outside of this "cone" would look distorted. This is what you have





to work with as a viewer.

When drawing a realistic view of an object or a scene, the artist must consider these rules of perspective before portraying the object on a flat surface. This flat surface, by the way (which for our purposes is the Mac screen), is called the "picture plane."

Consider the proverbial scene in which you're standing in the middle of a straight and narrow train track with telephone poles on either side of you and all points converging at a single point off in the distance. This point at which all the lines converge away from you is called the "vanishing point," and the horizontal line on which the vanishing point lies is called the "horizon line."

And if you happened to have a medium pane of rectangular glass with you and you held it out in front of your face and viewed the scene through it, this pane of glass would be the boundaries of your drawing. Depending on the viewer's eye level, line of sight, station point, and vanishing points in relation to the horizon line, the same object or scene could be viewed in many different ways.

A cube can be drawn in such a way as to describe the exact position of the viewer. The process used to draw it accurately may be labeled "one-point" or "two-point" perspective. You can have multiple vanishing points, multiple horizon lines, and "bird's-eye" and "worm's-eye" views, as well as different vanishing points for ascending and descending planes. But for what it's worth, just remember that all parallel lines converge on their own vanishing points on a horizon line.

The easiest way to prove some of these theories is by sitting down at the Mac and drawing a three-dimensional cube. Define the front surface of the cube by drawing a square using the open rectangle selection from the tool palette in *MacPaint*. (If you hold down the shift key as you draw, you'll produce a perfect square.) Now repeat that step, except this time draw the second square just behind the first square and slightly above or below and to one side of it. Make the second square a little smaller. Connect the points at the corners with the line tool, and after you clean up the construction lines with the eraser you have a cube. Following the examples will help you develop a sense of the rules firsthand.

How much you need to know about perspective is an individual choice. If you have an architectural bent and plan to use your Macintosh for floor plans and perspectives of interior or exterior projects, further research into the subject will be necessary. But for the rest of us, it's enough merely to take a look around and see the living examples of these perspective theories. Notice how the lines of a building converge to imaginary vanishing points and how when your eye level is lowered you can see more of bottom planes of objects and vice versa. You'll find there is no substitute for direct experience.

Scale and Proportion

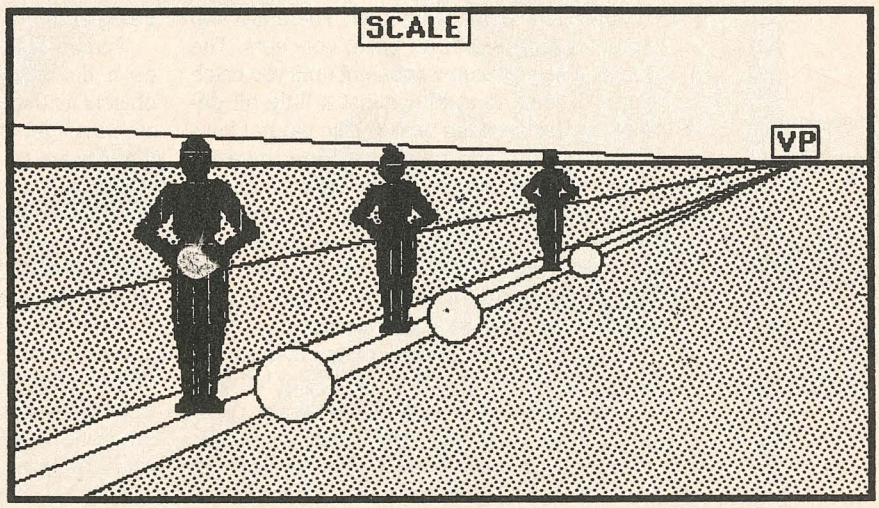
Scale and proportion are by-products of the laws of perspective and can be determined once you've established an object of a known and recognizable size that you can relate other things to. Starting a drawing with an object that is recognizable as being a certain size allows you to portray realistically in perspective any other objects you wish to make part of the scene. All you have to do is follow some very simple rules. To show the smaller parts of an object in correct proportion to its whole, you can fall back on the basic shape idea and "eye" the drawing into form. Soon this way of looking at things will become second nature to you and you'll be able to sketch things in quite accurate and proportional scale.

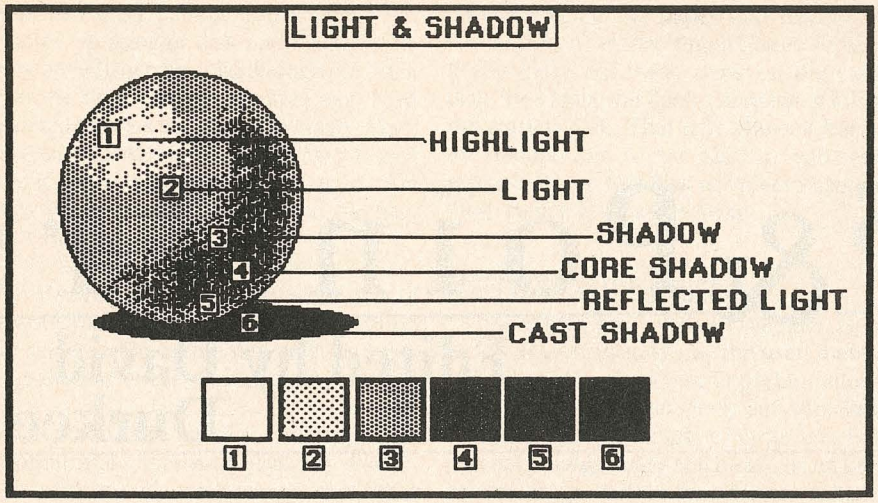
Light

Light surrounds us. It's what defines the dark. There would not be dark without light.

Light and shadow go hand in hand to breathe life into your drawings. No matter where your light source is located, it will travel in a straight line and strike any object that is not obscured from that source by any other objects.

In the traditional procedure for describing





form, called *chiaroscuro*, there are roughly six distinct tones of shading that describe a lit object. These are the highlight, the light, the shadow, the core of the shadow, the reflected light, and the cast shadow.

The "highlight" is where the light source hits the object. This can be a small white dot that describes a pinpoint light source, or it can be several dots indicating more than one light source. The "light" area is the area that fills the general surface that is facing the source, and the "shadow" is basically the area that is not in direct contact with the light source. The "core of the shadow" is the darkest area of the object, and the "reflected light" is the light bouncing back at the object from the surface it is sitting on. The "cast shadow" falls on the surface the object is on and describes the angle and strength of the light source.

MacPaint has several lighter and darker patterns; this lets you fill the areas to create a feeling of value and form. Experiment with these. The most ideal patterns for our purposes seem to be the ones with an even dot pattern. Choose the denser ones for darker values and the less dense ones for lighter values. Overhead lighting emphasizes the bulk or mass of your subject, while side lighting gives dramatic contrast and design because of its strong lights and shadows. The same is true of underlit scenes. We all remember the scary lighting used in horror movies where the monster is always shown lit from below. This lighting technique adds drama and suspense to an otherwise normal scene.

Robot Wars

Now here's an exercise to help demonstrate the ideas we've covered so far. It's time to create your own object with the help of *MacPaint*. You've seen the robot that neatly adorns the Scrapbook. Well, that's our inspiration here.

Your assignment is to create your own robot in any shape or design you want. Remember that form follows function, so decide what kind of activities you'll be asking your robot to perform and keep that in mind as you design it. You're the one who has to live with your creation, so make it something you want running around your home or workplace.

Here's how to begin. If you drew some basic shapes last month and you followed instructions and saved them to a file, open that file. If you didn't, draw some small circles, squares, triangles, and rectangles off to one side of the screen. From here you can use the selection rectangle and lasso from the tool palette to build your robot.

The selection rectangle and lasso are used to select an area from your drawing. Once the area you wish to manipulate is selected (enclosed in that wild animated outline), you can move it around. Remember, the cursor must change from its cross shape to a solid arrow before you can drag the object in the selection rectangle or lasso.

The object inside the selection rectangle may be inverted, flipped horizontally or vertically, or rotated using selections from the Edit menu. Lassoed objects may only be inverted.

You can make a copy of the selected object by choosing Copy and then Paste from the Edit menu, or you can duplicate the object from the keyboard by pressing the Option key and dragging the image. To make repeating copies, hold down both the Option and Command keys while you drag. Holding down the Command key while dragging the selection will stretch your picture up and down or left and right as if it were on a rubber sheet.

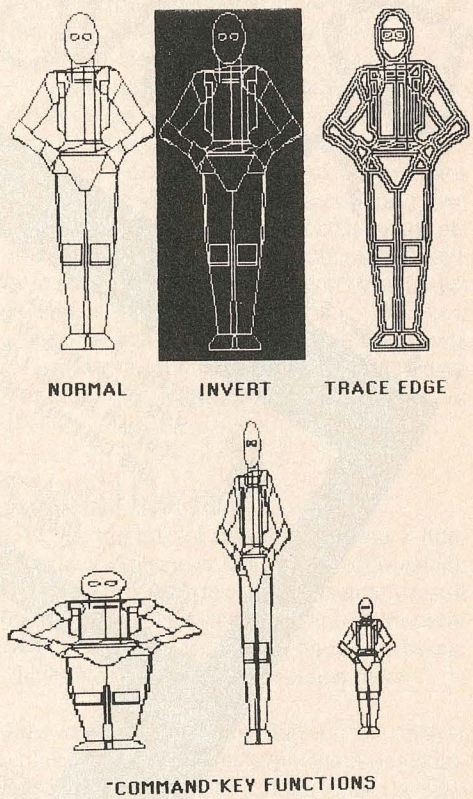
Use the lasso to outline irregular shapes, and use the techniques described above to stack the geometric shapes to create the body of your robot. Draw new shapes, like rectangular boxes and smaller circles. It's your robot—anything goes. You needn't shade the robot until later.

If the arms, legs, or other appendages on both sides of your robot are to be identical, you can save time by creating one appendage and then flipping it horizontally using the selection rectangle.

Once you're happy with your new drone, shade it by using all the patterns at your disposal, keeping in mind the location of your light source. Make the planes that would be facing the light the lightest in value. This can be done by filling those surfaces with the lightest pattern (usually the one with the fewest dots). Then go back in with a brush and take out some dots to simulate a highlight. Experiment with all the patterns to create different surface textures, always keeping in mind where your light source is located.

One word of advice: Whenever you get to a point where you like what's on the screen, save it as a file for later recovery. The Save As selection from the File menu will save the screen as a new file, prompting you for a new name. The Save selection will replace the old file with the new one under the old name.

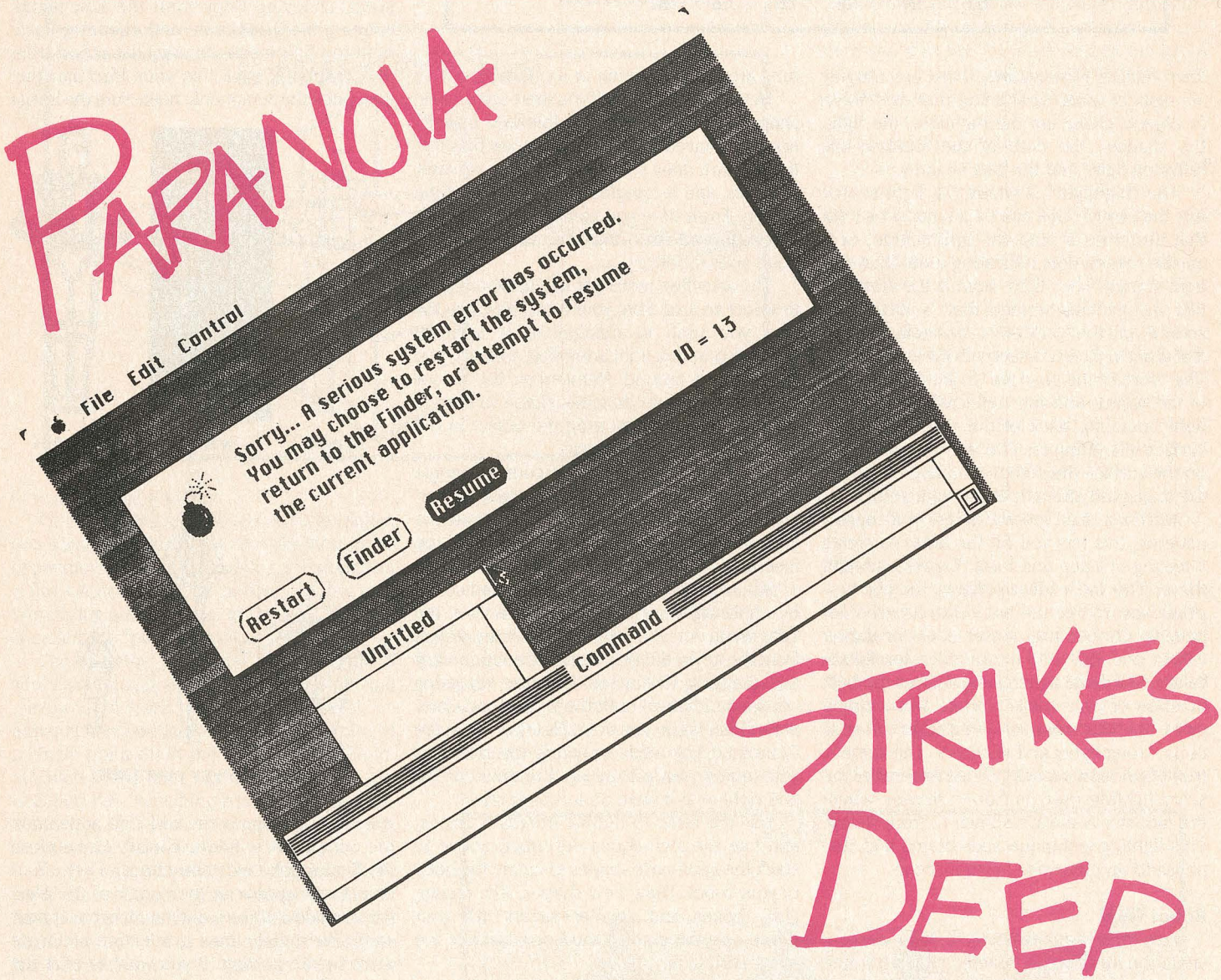
When you get your robot the way you want it, you may want to save it onto the Scrapbook by enclosing it within a selection rectangle or lasso, choosing Copy from the Edit menu, opening the Scrapbook, and choosing Paste from the Edit menu. To copy the object from the Scrapbook back into your *MacPaint* picture, open the Scrapbook, make sure the image you want to copy is the one that appears in the Scrapbook window, choose Copy, close the Scrapbook, and then choose Paste. Your object will appear in the center of the *MacPaint* window. If you want the object to appear somewhere else, draw a selection rectangle there before pasting. If you want to alter the shape of the picture, resize the selection rectangle before pasting.



We want to see the results of your efforts. So when you're through putting the finishing touches on your robot, why not print it out and send it to ST.Mac Robot Wars, Box 7041, North Hollywood, CA 91605. Make sure to include a note listing any particular tasks your robot was designed for. We'll print the most imaginative and visually appealing submissions in a future column. Have fun!

Clicks & Pointers

Edited by David Durkee



Bring 'Em Back Alive

One of the most unpleasant things to have happen when you boot up a Macintosh disk is to be greeted with an error message on-screen before you've even gotten started. The disk will not boot. It appears to have crashed. Trusted sources tell us that there is a way to recover from a crash on boot-up. And we're told that an unnamed Apple engineer claims this technique works nine times out of ten: While the disk is booting, hold down the Option and Command keys.

Eager to test the validity of this method we deliberately set out to crash a disk. Naturally, we worked with a backup copy. In the course of our efforts, we discovered a heretofore unstated corollary to Murphy's Law: The chances of a disk crashing are directly proportional to the importance of the data stored on that disk. Since our disk was a backup, it had no importance in and of itself; therefore the chances of our crashing it, even deliberately, were near zero.

Let that illustrate the importance of making copies of valuable disks. Murphy's Law is just perverse enough that it can occasionally be used against itself.

Despite the odds, we did manage to crash

the disk after about an hour. Here's how: First, we booted up another disk and ejected it by pushing an unbent paper clip through the small hole to the right of the disk drive slot. Next we inserted the disk we wanted to crash. When you exchange disks in this way, the Mac doesn't notice; it still thinks the first disk is in. The window on the screen still showed the files from the first disk.

So we deleted one of those files.

The Mac responded by deleting an area of the second disk corresponding to the location of the selected file on the first disk, and this had the desired effect of crashing the second disk.

When we tried to repair the damage with the Option and Command keys, we discovered that this type of crash can't be fixed that way. The point was conceded to Murphy.

Later on, we succeeded in crashing a disk in such a way that the prescribed method did work. The method is too tedious to relate here, but what it amounted to was damaging the Desktop file. The Desktop file (which is something like a file directory) is normally invisible and therefore rather difficult to damage intentionally.

The hallmark of a crashed disk that can be fixed in the manner our sources described is that it seems to boot until it gets to the point of displaying the disk icon on the screen. Then it stops. A disk that boots up to a black screen with a sad Mac face on it (as our first disk did when we crashed it) doesn't seem to respond to this treatment, however.

Disk Recycling

When attempting to recover a crashed disk try any means you can. Try the method described above. Try booting up one disk, ejecting it with the Eject command from the File menu, inserting the crashed disk, and copying files off of it. (If this method isn't going to work, the system will probably crash after the third step.) Try conducting a seance. If none of these strategies work, put the disk away and hope someone else has a better idea later on. If, however, you have lost all hope of data recovery and you find yourself badly in need of a blank disk, let go of your hopes and erase it.

Easier said than done.

Sometimes a disk just doesn't want to be erased. It won't boot, remember? And you can't eject a bootable disk and insert this one without it crashing—that is, if it knows what you're doing. As you'll recall, however, there is a way to swap disks without the system being the wiser. Use the old "unbent paper clip in the hole next to the disk drive" trick.

There's a catch to this method. You have to boot up one disk, eject that disk with File/Eject (the Eject command on the File menu), insert another *good* disk, eject *it* with the paper clip, insert the bad disk, and erase the bad disk with Special/Erase Disk (the Erase disk command on the Special menu). The double ejecting is necessary because, when a disk is replaced using the paper clip method, the Finder doesn't realize that it's been replaced. If you carry out these steps on the startup disk, the Finder will think you're trying to erase the startup disk. That isn't allowed because the Finder has to be able to get system resources from this disk in order to keep running. Hence the double swap.

MacPaint to the Rescue

No matter how well a computer is protected against mistakes by the user, there are a number of ways to lose important information. Electrical power goes out sometimes: Even if the power company in your area is 100 percent reliable, fuses still blow, circuit breakers still break, and runaway tractor trailer rigs still careen wildly into electrical lines. Occasionally. Or maybe you just bump the reset button with your elbow. And it always happens just as you're about to save the labors of the last three hours.

It can happen to you. If the file you're working on is in *MacPaint*, however, you have a chance of getting it back.

You see, the Mac can detect the power starting to dip microseconds before it goes below the critical level, and in that time it will save your entire document to disk.

You don't believe that?

Okay, here's the truth—first the view from outside the Mac, then the inside story.

Let's say you're working on a bit-mapped masterpiece in *MacPaint*. For whatever reason, the power goes off and you lose your work. Disgusted, you give up on being a computer artist and go read a good book.

One week later, you boot up *MacPaint* again, just to do some doodling. You notice two files on the disk called *Paint1* and *Paint2* represented by generic blank icons. Since neither of them will open when you double-click them ("An application can't be found to open this file"), you decide that they're some obscure pair of system files or something and superstitiously leave them alone, hoping that they'll go away. You double-click the *MacPaint* icon to begin your doodling.

When *MacPaint* comes up, you notice something odd. First of all, there is a picture displayed. Second, the title bar of the work area contains the cryptic word *rescue* instead of the expected *untitled*. Next, you observe that the picture being displayed is none other than the one you were working on the week before.

Unfortunately, when people discover this, it seems that what they were working on was usually a birthday or anniversary card, and by the time it's rescued the event has passed. This occurrence is covered by two corollaries to Murphy's Law: the one about lost data files and the one about lost causes.

Now, here's what happens inside the Mac. A full-size *MacPaint* document represents the 8-inch by 10-inch printable area of an 8 1/2-inch by 11-inch piece of paper. At seventy-two pixels per inch and eight pixels to the byte, this occupies about 50K of memory space uncompressed. Since this is more memory than can really be spared in a 128K Macintosh, *MacPaint* uses the disk to store the document once the file gets too large to fit in the working window all at once. That is, whenever you use the hand tool (the grabber) or the Show Page option to move the window, *MacPaint* goes to the disk and saves a compressed image of the picture as it exists. When you move the window back to an area you've already drawn on, *MacPaint* gets the image there from the disk.

Paint 1 and Paint 2 are the disk files in which *MacPaint* keeps the image. When you quit *MacPaint*, whether you've saved or not, these files are deleted. If the power goes down or you reset out of *MacPaint*, however, they aren't erased. The next time you run *MacPaint* (if you do so by opening the application rather than opening one of its other data files), it sees those files on the desktop, reads them into memory, and lets you know it has rescued the file you thought was gone forever.

It's not clear why *MacPaint* uses two files for this purpose. Both are essential, however; if you delete one, *MacPaint* can't rescue even part of your picture. The technique in evidence here does explain one of the mysteries of *MacPaint*. When you start out with as much as 30K free on the disk, you occasionally get the message, "The disk is almost full. Delete some documents or change disks," even though you haven't even saved anything. *MacPaint* needs enough disk space to keep its records of your work on disk and still be able to save the picture as a *MacPaint* file when you ask it to.

A Morbid Glossary

If all the talk of crashing has you a little reluctant to use your Mac for fear of breaking it, don't worry. Very few crashes actually result from damage to the computer, and we've never seen a crash cause the computer to break. Here's what all those crash terms mean:

File crash: One of the files on the disk—either a data file or an application—is damaged to the point that it can no longer be read or written to.

Disk crash: The same condition, but apparently affecting the entire disk, or at least its file directory, making the disk unbootable and all the files inaccessible.

System crash: Something happens to the application or operating system of the computer that causes it to freeze up and cease to respond in a normal way to user input. In the Macintosh, crashes of this sort usually result in an alert box offering the options of rebooting, occasionally returning to the Finder, and rarely continuing with the current application. In other words, the Mac has rather good error trapping.

Do you have a Macintosh or Lisa tip to share with other readers? If so, send it to Clicks & Pointers, Box 7041, North Hollywood, CA 91605. Contributors will be credited in the magazine.

TELECOMMU

By Matt Yuen

MacTerminal— Now the Fun Begins

Finally, finally, after much speculation in the minds of Macintosh owners, *MacTerminal* has arrived. It was pretty funny for a while, everybody cuddling up with their machines and no software to connect their modems to the outside world.

Anyway, *MacTerminal* is here, and, to no one's surprise, it works very much the way *LisaTerminal* does. If you haven't already done so, go dig out last month's copy of *ST.Mac* and read the stuff in Telecommunications about *LisaTerminal*. If you don't have a copy of the issue (or if you already tore out those pages to house-train the dog), well, photocopies are cheap.

For people who were blown away by the complexity of terminal programs on IBM, Apple, Kaypro, and other microcomputers, *MacTerminal* will be a relief. But if you're one of those who eventually became proficient at using those programs, you're likely to find *MacTerminal* confusing and slow. Nonetheless, it's worth learning, since it's *the* communications program. For now, anyway.

MacTerminal lets you receive stuff from other computers and send stuff to other computers. When receiving text files, you can capture the information that comes into the Mac and then put it on a disk as a document that can later be loaded into *MacWrite* for editing and formatting. Conversely, anything you can create on *MacWrite* can be sent to another computer as text. When communicating with another Mac, you can even transfer programs and *MacPaint* files.

Let's find out a little more about the *MacTerminal* program.

Hors D'oeuvres and Entrees

The hardest part about *MacTerminal* is remembering where everything is. The first thing to do when you get the program is to play around with the various pull-down menus. *MacTerminal* has five menus; in addition to the usual File and Edit menus, it offers Command, Settings, and Phone. Command and Phone aren't too hard to figure out; the real meat of *MacTerminal* is in the Settings menu.

The Settings menu isn't that different from *LisaTerminal*'s Setup menu. Here's what's on it:

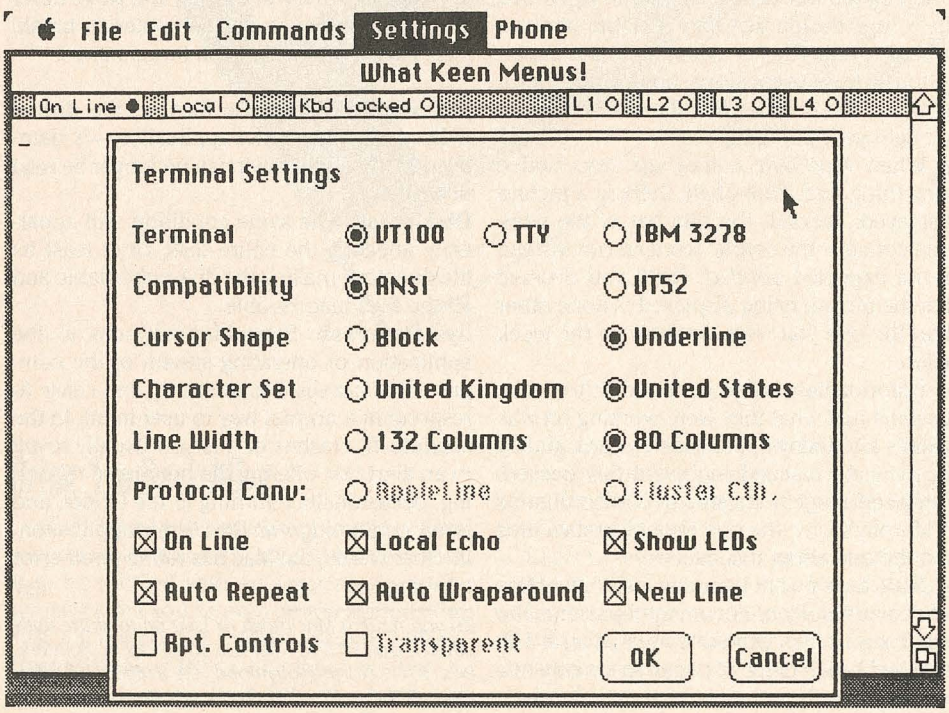
Terminal. *MacTerminal* knows how to emulate (make the Mac look and act like) two of the most popular mainframe terminals: the Digital Equipment Corporation (DEC) VT100 and the all-purpose Teletype, or TTY, terminal. The Terminal dialog box lets you customize the way the Mac displays things on the screen. You get your choice of terminals to emulate (VT100, TTY, IBM 3278), a block or underline cursor, the United States or United Kingdom character set (the only difference is that the U.K. set displays the sign for the English monetary pound instead of the # sign), a line width of 80 or 132 characters, a local echo (half or full duplex), an automatic word wraparound, and some other things we don't have to worry about right now.

When you open *MacTerminal*, the terminal settings are already at those that Apple considers to be the most commonly used.

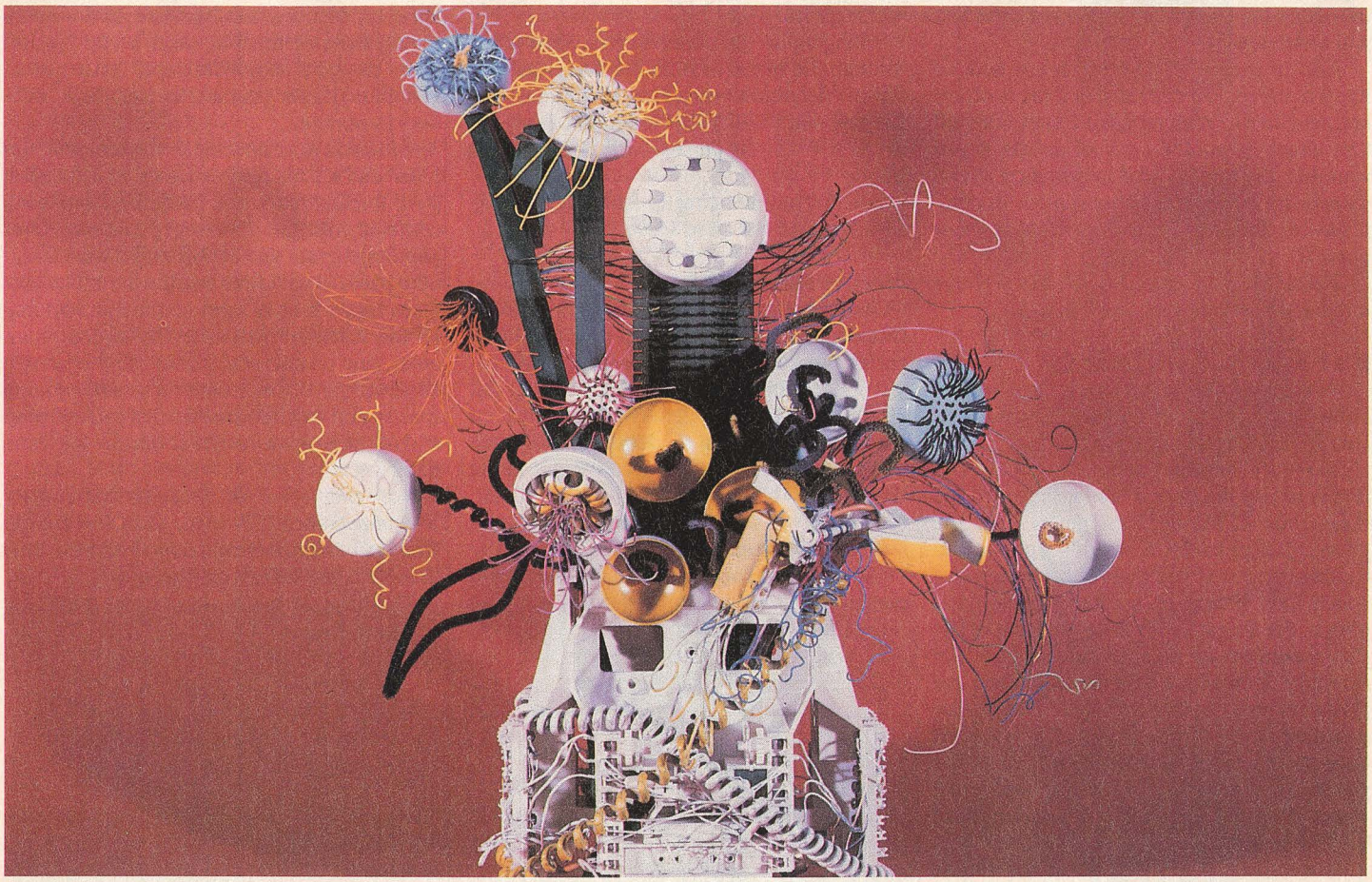
Tabs. These things work just like they do in *MacWrite*. Tabs are already set at every eight spaces. If that's too many, then yank off a few; if it's not enough, then pick some up from the tab well and slap down as many as you want, where you want.

Communication. This box is similar to the Computer Compatibility box in *LisaTerminal*. You can choose any of the baud rates from 50 to 19200 (yes, 19200), but the most commonly used rates are 300 and 1200. Also in this box are choices of seven or eight bits per character; even, odd, or no parity; and xon/xoff or no handshaking.

Answerback Message. Don't worry about



NICATIONS



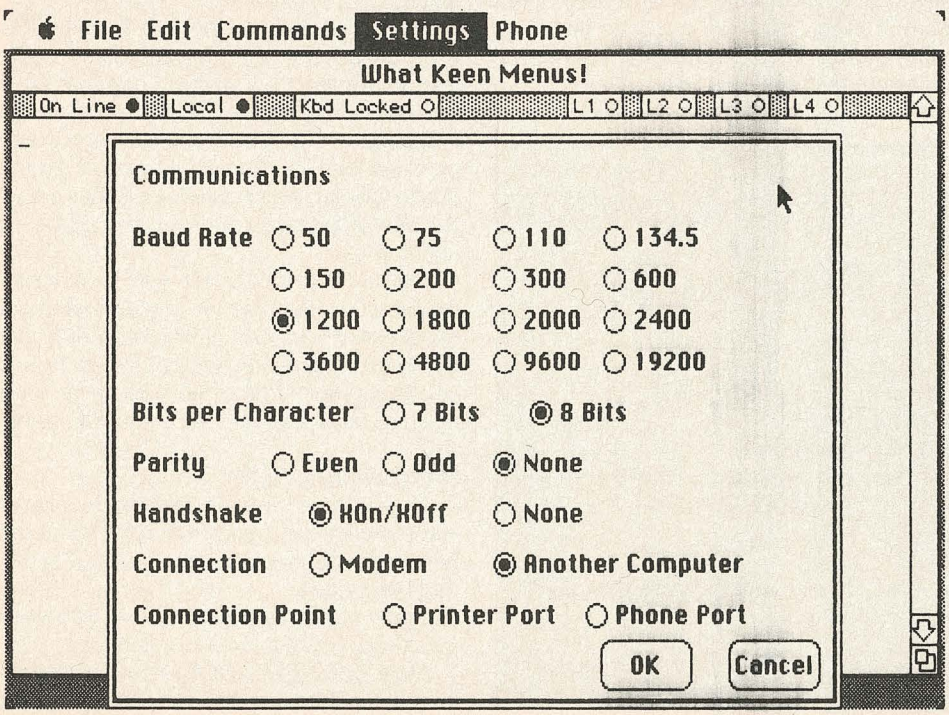
this right now. It just lets you type in whatever message you want to have appear on the screen when somebody else's computer calls yours.

File Transfer. Transferring files is a little tricky the first few times around. *MacTerminal* offers two ways to do it: text or modem7. Here's the difference:

Let's say you've written the Preamble to the Constitution on *MacWrite*, and now you want to send it to another computer. If the other computer is a Mac, you can simply send the *MacWrite* file (modem7 method). If the other computer isn't a Mac, you can send the *text* of the file ("We, the people...") but not the file itself. Confused? It's like the difference between mailing a book to a friend and mailing the book's words written down on paper.

Modem7 is a *protocol* for, or method of, transferring information. When you use modem7, you can transfer programs, *MacPaint* files, and *MacWrite* files, and they'll all arrive at the other computer in flawless condition. That's because modem7 has a way of checking to make sure that when information arrives at the other computer, it arrives exactly as it was sent; if there's any discrepancy (for example, if you send *friend*, and it arrives as *fiend*), the information is sent again. The only catch is that the computer you're sending to has to be a Mac that's also running *MacTerminal*. Well, not exactly.

When you have two Macs running *MacTerminal* on each end, it's possible to send files back and forth error-free. Modem7 is based upon something called the Christensen protocol (or xmodem, famous in computer hobbyist circles), which implies that it should be possible to exchange files between a Mac



armed with *MacTerminal* and another computer running a program that uses the xmodem protocol. Even if the two computers are completely dissimilar, it's possible to transfer a *MacWrite* file in text form (rather than sending it via modem7), but noisy phone lines introduce the possibility that not all the text will arrive intact.

Also in the File Transfer box are options for what the remote computer is (Mac or other) and whether to remember or forget lines off top, which is another way of asking if you want *MacTerminal* to hold information after it disappears off the top of the screen. In computerese: "Do you want the buffer open or closed?"

Phone. This is where you type in the phone number you're going to call.

If a Person Answers, Hang Up

With all that out of the way, let's call somebody. From this point on, *MacTerminal* and *LisaTerminal* operate very similarly.

There are two ways to dial the phone. The first is to use the Phone menu. In the Phone dialog box in the Settings menu, enter the phone number to call. Some office phone systems require that the dialing be done on a Touch-Tone phone or a rotary (pulse) dial phone. For Touch-Tone, put a *t* in front of the phone number; for pulse dialing, use a *p*. For example, Touch-Tone callers would enter the phone number like this: t555-3574. Once you've specified a number as either t or p, you don't have to do so again unless you switch off the power to the modem. (For *LisaTerminal*, you'd select the Enter Phone Number option from the Phone menu.)

Once that's done, pull down the Phone menu and click the Dial option. On the screen you'll see something that looks like this:

ATDT 555-3574

That's because the command that tells the Apple modem to dial the phone is "ATD." The letters AT stand for *attention*, and D stands for *dial*. The T after the D, in our example, stands for *Touch-Tone*. For pulse dialing, it would have been ATDP 555-3574.

Using the Phone menu to dial is fine for keyboard klutzes, but every time a menu is opened, the program stops and goes to the *MacTerminal* disk. If you find that too annoying, you can always type the dial command at the keyboard (ATD 555-3574), and the modem will respond.

Hanging up the phone is also done either through the menu or from the keyboard. The Phone menu provides a Hangup option; from the keyboard, typing ATH (attention: hang up) or + + + (three pluses) will hang up the phone.

Dialing and hanging up isn't very exciting, so onward to the good stuff.

Practice Makes Perfict

As easy to use as the Lisa and Mac are supposed to be, learning to use terminal programs always takes a bit of practice. The easiest way to practice receiving data is to call up a bulletin board system (BBS) or to call up someone else with a modem and type back and forth to one another. Don't forget to set "Remember Lines Off Top"; if you forget, everything will be dumped into oblivion once it disappears from the screen.

After hanging up, you can review what you received by scrolling up and down with the vertical scroll bar on the right side of the screen. Save the information to disk by selecting Save or Save As from the File menu. When you view the disk contents, the file will appear as a *MacTerminal* (or *LisaTerminal*) file, just waiting for you to load it into a word processing program.

Sending text information is as easy as receiving it, but getting everything set up takes a bit longer. To create a text file for sending, you can use either a word processor or the terminal program. Using the terminal program is fine for small and simple messages, but if you blow it and have to edit the message, you're out of luck. The terminal program doesn't let you move the cursor around; once a line scrolls up, you can't touch it except to copy it to the Clipboard.

Though it's easier to create messages and text files in *Write* than in *Terminal*, sending such files is more complicated. First, select all of the document, either from the Edit menu or with the cursor, and copy it to the Clipboard. Next, pull out the terminal program. (For this step, Lisa owners have the upper hand on Mac owners because of the Lisa's multitasking ability. Mac takes seemingly forever to quit *MacWrite* and open *MacTerminal*.) Enter phone number. Dial. Connect to remote.

After you're connected to the remote computer and it's all set to receive your file, just go to the Edit menu and paste the Clipboard's contents into the *Terminal* document. Your computer treats this action as if it were a regular paste command, but the terminal program treats it as a send command, sending all the text information to the remote computer, one character at a time. To the remote computer, it doesn't matter if you're sending (pasting in) text information or if you're typing real fast; it sees the results of both as the same thing—text.

For the sake of traditionalists, *MacTerminal* and *LisaTerminal* send text no differently than most terminal programs do; only the words are different. Instead of

text file —> buffer —> remote computer

we have

Write file —> Clipboard —> *Terminal* document.

Things To Do during a File Transfer

Sending *MacPaint*, MBasic, and other kinds of files is a lot simpler—none of this copy-to-Clipboard nonsense. Instead, just open the File Transfer dialog box in the Settings menu and select the Modem7 option. Then go over to the File menu and click the Send File option, which will display the names of all files on the disk. Pick a file, click the Open option, and then go get a soda while the transfer takes place.

(Fans of *LisaTerminal* can't transfer files this way, because the program handles only text information. *LisaWrite*, *LisaCalc*, and *LisaGraph* files can be transferred, but that's it. When, or if, Apple or other software developers will decide to make it possible for the Lisa to transfer nontext files is anybody's guess. For now, all we can do is sit here and stew about it.

It'll be interesting to see how other terminal programs that use the xmodem protocol handle *MacPaint* files or *MacWrite* files that contain various fonts. Receiving such files is one thing, but what to do with them on a machine other than the Mac is something else.

Problems and Fixits

Now that we have the basic knowledge of *MacTerminal* and *LisaTerminal*, let's find out how to fix various problems that can occur. And they will.

Problem: Can't see a darn thing I type when I'm connected to a remote system.

Fixit: Lisa: Open the Computer Compatibility box in the Setup menu and change duplex

Some Elementary Apple Modem (Hayes Modem) Commands

Command:	Means:
AT	Attention: modem, the following is a command.
ATDT xxx-xxxx	Touch-Tone dial the number xxx-xxxx.
ATDP xxx-xxxx	Pulse (rotary) dial the number xxx-xxxx.
ATD xxx-xxxx	Dial the number xxx-xxxx. You can use this form without the T or P if you used T or P in an earlier ATD command.
ATH	Hang up the darn phone.
+ + +	Hang up the darn phone.
A/ (no return)	Repeat the previous command. This comes in handy for quick redialing.
AT M0	Shut up; turn off the speaker.
AT M1	Speak up; turn on the speaker.

(under Terminal) to half. Mac: Open the Terminal box in the Settings menu and put an X in the box by Local Echo.

Problem: Every character I type appears double on the screen.

Fixit: Lisa: Change duplex to full. Mac: Switch Local Echo off.

Problem: Pressing the return key puts the cursor at the beginning of the line I just typed. Also, incoming text appears on top of the line that came in before it.

Fixit: In the Terminal dialog box, click Auto New Line to on. This puts the cursor on the next line down when the return key is hit or when incoming text wants to start a new line.

Problem: All the characters (typed and incoming) pile up on each other at the end of the line.

Fixit: Click Wraparound on (under Comfort for Lisa, Terminal for Mac).

Problem: The stuff that comes from the remote computer looks like garbage.

Fixit: Check the baud rate. If both computers' baud rates were the same when the connection was made, then changing the baud rate should take care of the problem. But if you were set at, say, 1200 baud when you called, and the remote was set at 300 baud, then you'll have to change the baud rate and call back. *Terminal* can't change the modem's baud rate "on the fly."

Problem: Can't find all the stuff I received. Vertical scrolling doesn't work.

Fixit: Switch on the Remember Lines Off Top option (Lisa: in the Comfort Settings box; Mac: in the File Transfer box). If lines are disappearing off the top of the window, and the vertical scroll bar on the right is white, then *Terminal* isn't holding the information after it goes off the top. A gray scroll bar with the little elevator box at the bottom of it indicates that Remember Lines Off Top is working.

If you just finished receiving a whole bunch of text and discovered that Remember Lines Off Top wasn't on during the transfer, then you'll have to receive the file all over again. Much to the phone company's delight.

Problem: Nothing appears on the screen except for little black rectangles.

Fixit: The parity is set wrong. There are only three settings for parity: even, odd, and none. Experiment until those stupid rectangles are gone. Change *LisaTerminal*'s parity in the Computer Compatibility box; *MacTerminal*'s is found in the Communication Settings box.

Words To Know

Baud. A unit of measure for the speed of data communication. Sometimes defined as the number of bits per second (bps) that travel between computers. Examples: 300 baud, 1200 baud, 9600 baud.

Bit. The smallest piece of information a computer can handle. In communications, one text character is represented by one byte, which can be made of seven or eight bits.

Byte. A set of bits that makes up an item of information, such as the letter A.

Full duplex. This just describes a situation in which information can travel in both directions between two computers at the same time.

Half duplex. You guessed it; in this case, information can travel in just one direction at a time.

Handshaking. This can mean different things. For our purposes, it's the ability to control the flow of data between the Lisa or Mac and the remote computer. When the buffer (lines off top) becomes full, the receiving computer transmits a signal telling the other computer to stop sending information until the buffer's contents is either cleared or saved to a disk. When the computer is ready to receive more, it transmits another signal to tell the remote that it's okay to continue sending.

New line. When this is switched on, the cursor goes to the beginning of the next line down when it receives a carriage return from your keyboard or the remote. When it's switched off, the cursor goes to the beginning of the same line it's currently on, and new characters are typed over old ones.

Noise. Anything that causes data to be transferred incorrectly. Static on the phone lines is an example of noise.

On-line. A connection between two terminals. When a computer isn't connected to a remote source, it's off-line. Off-line is sometimes referred to as local, meaning you're communicating with yourself. Characters appear on the screen, but they're not going to another computer.

Parity. The process of checking that data was transferred without error. A parity bit is either a 1 or a 0. The sending computer adds the parity bit to the end of a byte, making the total number of bits an even number (if parity is set to even) or an odd number (if parity is set to odd). The receiving computer inspects the byte it received to see if the number of bits it contains is the same (odd or even) as when it was sent. Selecting no parity ignores the whole messy process.

Protocol. A set of rules governing the format and control of data transfer. In English: Using a protocol to send data makes sure that everything is transferred in perfect condition (no characters are lost or changed, and no extra information is added). There are several protocols in the communications world; the sending and receiving computers must be using the same protocol for error-free transfer to occur.

Wraparound. This means that when the cursor gets to the end of the line, it automatically goes to the beginning of the next line down. If wraparound is switched off, characters stack up on each other at the far-right side of the screen until the computer receives a carriage return.

You Chose the Right Computer Now Get the Right Carrying Case

MacPak»

The ideal campus or travel case. Heavy divider separates accessory pockets for papers, manuals, ten-key and room for an extra disk drive & modem. Cordura® & Masonite, a thick high-density foam make this quality pack shock and weather resistant. It can be carried by shoulder sling, like a suitcase or by hideaway backpack straps.
Black, Navy Blue, Grey, Burgundy.
$79.00 (add $5.00 for shipping)

MacBag

Constructed the same as the MacPak, but without the backpack strap.
Black, Navy Blue, Grey, Burgundy.
$69.00 (add $5.00 for shipping)

The Best Quality Pack on the Market Today

MacFreighter

A heavy duty case designed for shipping as cargo or checking as baggage. Heavy shipping foam throughout. Can be easily carried suitcase style by one person.
$199.00 (add $10.00 for shipping)

Print·Pak

Cordura® and Masonite make this case essential for transporting the Imagewriter® printer. Can be carried by shoulder sling, like a suitcase or attached to the MacPak while carried as a backpack.
Black, Navy Blue, Grey, Burgundy.
$49.00 (add $3.00 for shipping)

MacCover

A beautiful vinyl leather, cloth-lined dust cover for the Macintosh,® keyboard, mouse & Imagewriter® printer. Aids in static prevention and kid protection.

Black or Brown.
$29.00 (add $3.00 for shipping)

To order call **1-800-227-3800, ext. 241**, or mail check, money order, MC or VISA to:

MacPacks,
P.O. Box 3928 • Kent, WA 98032
(Washington residents add 7% sales tax)

MacPacks

Carriers, Packs and Covers for the Macintosh®

For more information call:
(206) 839-0432
IMMEDIATE SHIPPING

Satisfaction Guaranteed. If for any reason, you are not satisfied with any item purchased, you may return it within 30 days for a complete refund. Rips or tears: send us the bag we will repair.

68000 Questions

By Andy Hertzfeld

Q: When, if ever, will a native-code Pascal compiler be available for the Mac?

A: Currently, Apple supports a native-code Pascal cross-compiler for Macintosh that is part of the Lisa Workshop, but right now it runs only on a Lisa with a hard disk. Think Technologies, the company that produced the interpreted Macintosh Pascal, is investigating the possibility of providing a native-code option in its developer's version of Mac Pascal, which won't be ready until the fall.

Q: Erase Disk does not work for a System Disk (it tells me I am not able to erase the System Disk). Is there any way to erase a System Disk?

A: Erase Disk will always prevent you from erasing the startup disk, so you may use Erase Disk to erase a System Disk as long as it is not the disk you booted up on. Simply boot up on the *Write/Paint* disk, or some other application, and then eject it and insert the System Disk you wish to erase. The Erase Disk command should now work.

Q: How will a hard disk drive be connected to a Mac?

A: There are a few different ways to connect a hard disk to a Macintosh. The most common way is to use one of the serial ports in a dedicated fashion. This allows data transfer rates of up to almost one megabit per second. If the hardware manufacturer follows our AppleBus protocols, you will be able to access the disk without dedicating one of your precious serial ports. AppleBus would also allow the disk to be shared by more than one Macintosh.

Another interesting way to connect a single-user hard disk to a Mac is by going through the external disk connector. This allows data transfer rates approaching 500 K bits per second and has the advantage of not tying up one of the serial ports.

Q: What will be Apple's policy for upgrading 128K machines to have more memory?

A: When the higher-density memory chips become available in sufficient quantities, Apple plans to offer a larger memory version of Macintosh. Customers who purchased 128K Macs will be able to upgrade to a larger memory configuration by returning their Macs to their dealer and purchasing the new memory. The old digital board will be swapped for a new one containing 512K of memory. This way, the new board will have been fully tested and burned in at the factory to ensure that you receive a reliable computer.

Q: What is contained in the System file and how much space does each component part take?

A: The data portion of the System file contains the RAM-based operating system, which is loaded into the system heap during the booting process. It is fairly small (about 2 to 3K) and consists mainly of patches to fix bugs in the ROM and some code to implement a few features that were invented after the ROM was frozen.

Most of the interesting stuff is in the Resource portion of the System file. This includes fonts; desk accessories; definition functions for windows, menus, and controls; packages like floating point numerics and disk formatting; some cursors, icons, and dialogs; and keyboard, printing, and international configuration tables. There are twenty-two different fonts that take up about 85K, or more than 60 percent of the System file. The seven desk accessories take up about 22K; the Puzzle is the smallest accessory at under 1K, while the Control Panel is by far the biggest at about 8K. The six different packages take up almost 16K, while the five definition functions occupy 4.6K.

Many people complain that there's not enough space on the *Write/Paint* disk. For the second release of the System (available this month), Apple decided to remove about half the fonts from the System file to free up some space. They are still available in the Fonts file on the System Disk, but you have to install them with FontMover before you can use them.

Q: What are the "virtual slots" I read about in a magazine article? How can I find out how to design a peripheral to use them?

A: There is really no such thing as a "virtual slot," but the article you read was probably referring to AppleBus, our multidrop peripheral bus. AppleBus provides an inexpensive way to connect as many as thirty-two different peripherals and other computers to your Macintosh. This allows people in the same office to share expensive peripherals like printers and hard disk drives. You can obtain complete AppleBus specifications from Apple Technical Support.

Q: How can I obtain a copy of the QuickDraw Programmer's Guide?

A: The QuickDraw Programmer's Guide is an important chapter in the *Inside Macintosh* book, available from Apple Technical Support. The Macintosh Pascal manual also has an excellent description of all the QuickDraw calls; that is scheduled to be available by the end of June.

Got a question? We'll try to answer it in this column. Send queries to 68000 Questions, Box 7041, North Hollywood, CA 91605.

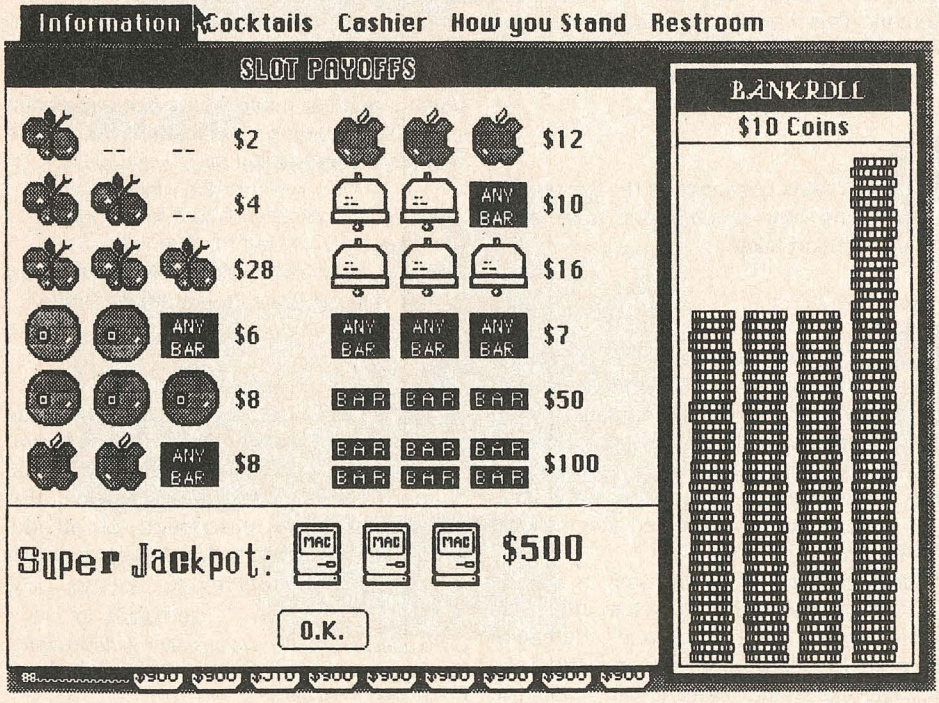
Mac-Slots

Soft-Life Gambles on Macintosh

By Steve Shendelman

Many of us are potential high rollers, especially when we're playing with pebbles, M&Ms, chips, or other imaginary tender. Gambling in Las Vegas or Reno or Atlantic City is the real thing, to be sure, but in those towns they insist on using real money. Apple owners have always enjoyed the alternative of having their computers run their own personal casinos, right down to friendly credit managers who let them get in over their heads and never send anyone to collect. Now Macintosh owners can enjoy many of these same thrills with a new package from Soft-Life called *Mac-Slots*.

Two games come on the disk, Keno and Slot Machine. When you insert the disk in the Mac, the scoreboard and title page appear. The scoreboard shows the top ten scores to date. Icons for the two games appear at the top of the scoreboard, allowing you to use the mouse and double-click to choose the game you wish to play.



Slot Machine is the more interesting of the two games. When you select the slot machine icon, a faithful replica of your basic one-armed bandit is drawn on the screen. Next to this is an area labeled Bankroll, which is filled with stacks of coins. Playing is as simple as using the mouse to grab a coin or coins, depending on your limit, moving them to the coin slot, and dropping them in. Then, again using the mouse, you simply grab the handle at the top, pull down, and release the lever. All three reels spin and slowly come to a halt. Payoffs are determined by the various combinations of symbols that appear. A Macintosh symbol is included, along with the usual cherries, oranges, and bars one finds on a typical slot machine.

Keno utilizes most of the same features for its play. The object of Keno is to match up to ten numbers of your choice with the twenty numbers randomly picked by the Mac. Payoffs are based on the amount bet and the number of correct choices.

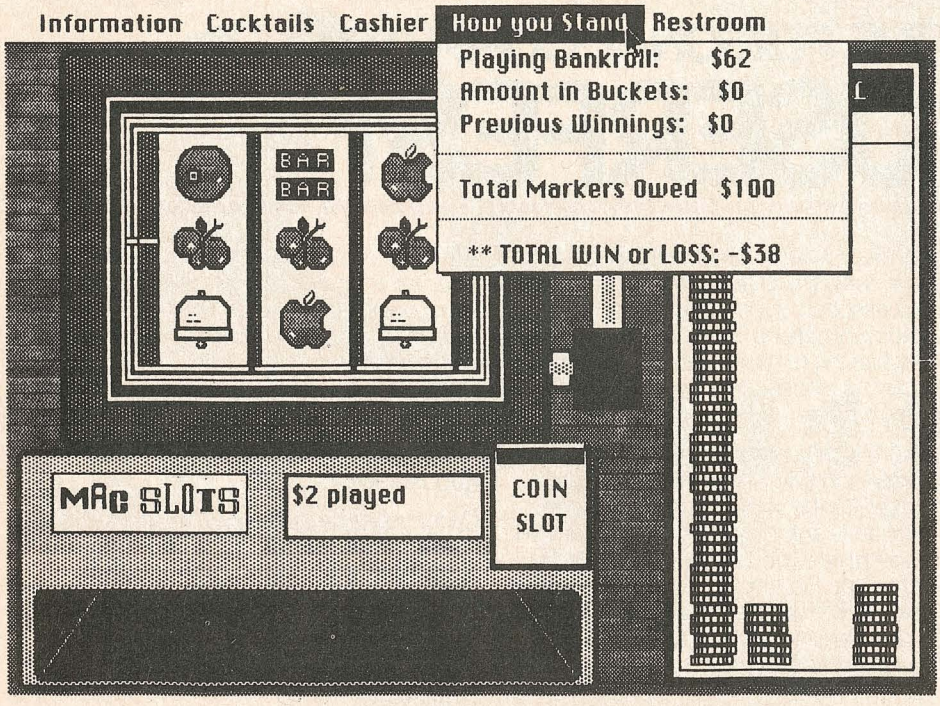
There are other features that both games have in common. These items are shown on the menu bar and are available at any time during play. The Cashier feature will change the value of your chips, extend credit, or let you cash in your chips (one way or another, depending on your luck!). The one labeled Cocktails seems somewhat bizarre at first, since a wet bar doesn't appear to be included in the Mac. This selection is used to choose betting limits, which increase as play continues. Selecting Information allows you to review the

payoff combinations and the house rules. Restroom might also be called Playing at Work; it enables you to silence the Mac or blacken the screen entirely. Sound and picture are restored by hitting return.

Soft-Life is also sponsoring a contest that all purchasers are automatically eligible to win. High scores (verified by screen printouts) will be published monthly in the company's ads. The contest will conclude November 30, 1984. The top ten scorers will win prizes, such as a Macintosh with Imagewriter for the highest score.

The games are nicely presented and play quite realistically, although one might have some misgivings about a slot machine that allows players to wager up to $40 at a time. The graphics are good and the package is easy to learn—almost too easy. And this brings up the hard part. The package sells for $77.77 (get it?). This is, of course, one of the first published games for the Mac, and therefore something of a novelty. But given that many of the Macintosh applications programs will sell for under $100, this seems a tad overpriced. Had this package included another casino game or two, like blackjack, poker, or roulette, the price might be more in line. Or, how about a "Slot Machine Construction Set" where you could change the parameters of the machine, such as the payoff combinations, the odds, or even the symbols on the reels?

Given the unknown make-up of the Macintosh owners (businesspeople versus college students), it is difficult to say whether the price will be a factor in the game's success. It's a good bet that a number of potential players would be a lot more willing to gamble on this product for somewhat smaller stakes.



Mac-Slots
Soft-Life
2950 Los Feliz Boulevard
Los Angeles, CA 90039
(213) 660-7940, ext. 561
$77.77

FEATURING IBM, APPLE & WORK-ALIKE COMPUTERS & COMPATIBLES

FORMERLY APPLEFEST & PC'83

THE PERSONAL COMPUTER USERFEST

Plan now to attend the Personal Computer Userfests, the largest events ever . . . for Apple and IBM PC users.

Userfest brings together two of the largest, most successful shows ever conceived for personal computer users: Applefest and PC'83. Now that Apple and IBM can run each other's software, and with so many products adapted for both systems, the two shows merged beautifully.

At Userfest you'll see—and try out—all of the newest state-of-the-art products for your Apple, IBM PC or work-alike. Each Show has hundreds of displays and exhibits, and thousands and thousands of products including innovative new software, power peripherals, accessories, support services, books and publications. Products to help you explore the full potential of your computer for office, home and school applications.

Userfest features all the major makes of Apple and IBM computer compatibles. In fact, it's the largest display of these products, and biggest gathering of IBM and Apple experts, ever assembled in either city. Hence, you can learn more in two days at Userfest than you could in months of visiting computer stores or reading trade journals.

And best of all, everything on display at Userfest is for sale, usually at special show prices, so you can save hundreds, even thousands of dollars by making your purchases at the Show.

So don't miss the Personal Computer Userfest when it comes to Chicago and New York in 1984. It's a once-only opportunity.

Order your tickets in advance and avoid long lines. Admission is $10.00 for a one-day ticket, or $20.00 for four days. Children's tickets (under 10 years of age) are $4.00 and $8.00. If you need hotel accommodations and/or airline reservations, check the line on the Advance Ticket form.

CHICAGO

USERFEST/CHICAGO
Thursday-Sunday
May 3-6, 1984
10:00AM to 5:00PM daily
O'Hare Exposition Center
9291 West Bryn Mawr Rosemont, Illinois
(next to Chicago's O'Hare Airport)

NEW YORK

USERFEST/NEW YORK
Thursday-Sunday
September 20-23, 1984
Madison Square Garden
10:00AM to 5:00PM daily

For information about exhibiting at the Personal Computer Userfests, call or write Northeast Expositions, 822 Boylston Street, Chestnut Hill, Mass 02167. Tel: 617-739-2000.

For hotel information call or write Trade Show Department, Fox Travel, P.O. Box 498, Waltham, Mass 02254. Tel: 617-890-1770 or 800-225-8410 ext. 314.

Userfest (formerly known as Applefest and PC'83) is produced by Northeast Expositions, 822 Boylston Street, Chestnut Hill, Mass 02167.

ADVANCE TICKET ORDER FORM

Mail this form (or a facsimile) with full payment to Northeast Expositions, 822 Boylston Street, Chestnut Hill, Mass 02167. Tel: 617-739-2000. No ticket orders accepted 14 days or nearer to each Show. Your tickets will be mailed one month prior to the Show. Sorry, no telephone or credit card orders please.

Name:_____

Company (if any):_____

Address:_____

City:_____ State:_____ Zip:_____

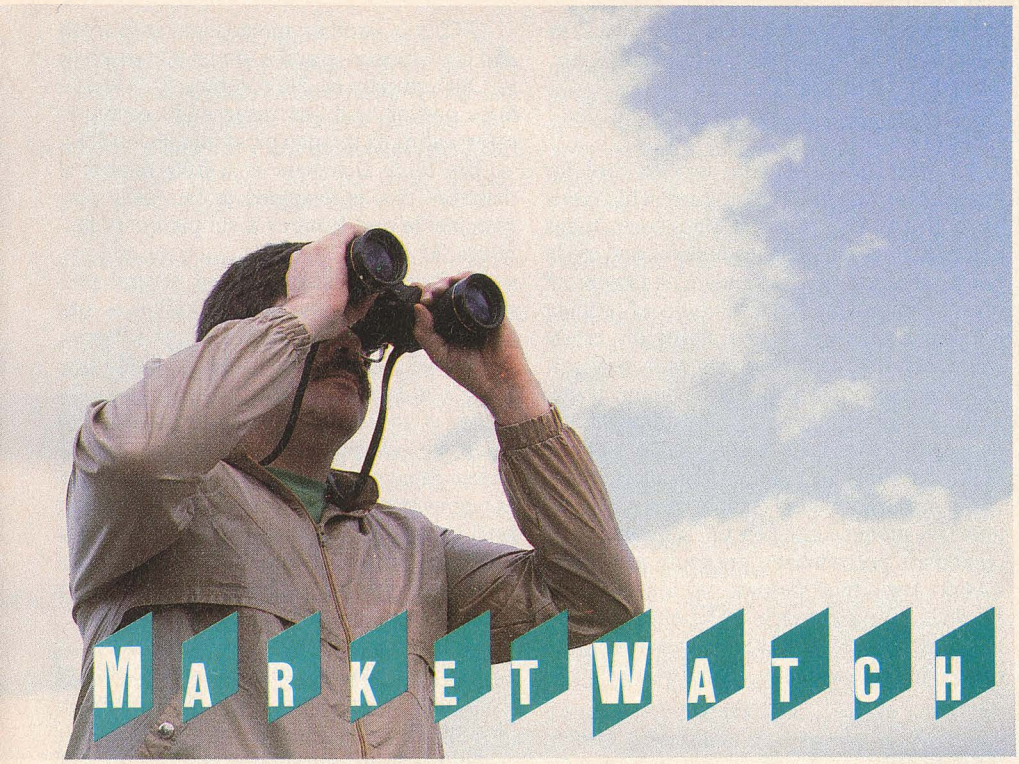
Tel: Day (_____)_____ Evening (_____)_____

Enclosed is full payment for:

____ quantity adult one-day tickets @ $10.00 each ____ quantity adult four-day tickets @ $20.00 each

_____
use this line for children's ticket orders

☐ Check here if you need hotel and/or airline information



Idea Processor

ThinkTank for the Macintosh allows planners to see their ideas in outline form. Users can collapse their outlines to see the big picture or expand them to reveal hidden details. *ThinkTank* is compatible with many other applications, and the software's outlines can be loaded into a word processor for formatting. $130. Living Videotext, 1000 Elwell Court, Suite 232, Palo Alto, CA 94303; (415) 964-6300.

Home Finance Manager

The Home Accountant lets users track any number of checkbook accounts, and print names and addresses directly onto checks; it provides for twenty-five automatic monthly transactions and flags items needed at tax time. *The Home Accountant* also enables users to enter monthly budgets for assets, credit card liabilities, and income and expense categories. Budget/actuals comparisons can be illustrated by bar, line, or trend analysis graphs. Reports can be composed as personal balance sheets, income and expense summaries (including net worth), budget/actuals listings, and various transaction activity reports. The program also includes a financial calculations module that allows users to calculate loans or determine the future value of monthly investments. $99.95. Continental Software, 11223 South Hindry Avenue, Los Angeles, CA 90045; (213) 417-8031.

Odesta Database

Odesta Helix is an interactive information system and relational database that takes advantage of Macintosh windowing and icon-oriented menu capabilities. The program features icon-defined relationships and calculations, as well as a user-customizable reports facility, query-by-form facility, and a feature that lets users cut and paste between forms. $395. Odesta, 3186 Doolittle Drive, Northbrook, IL 60062; (800) 323-5423.

MegaMerge

MegaMerge is a Macintosh mail-merging program that works with *MacWrite* documents and enables users to create form letters, print mailing lists, address envelopes, and allows long documents to be merged together so that they can be printed as one file. $125. Megahaus, 5703 Oberlin Drive, San Diego, CA 92121; (619) 450-1230.

Interactive Detective Story

CBS Software is releasing *Mystery Master: Murder by the Dozen* for the Macintosh. The program contains twelve murder mysteries that can be solved by one to four players who compete to find the guilty person as well as the motive. Time is a factor in each game and helps determine a detective's rating if he succeeds in solving the mystery. Players start with a description of the details of the scene of the crime and are given choices of where to go and whom to question first. Players' questions are rewarded with numbered clues that may include red herrings. $34.95. CBS Software, One Fawcett Place, Greenwich, CT 06836; (203) 622-2615.

Statpro

Statpro: The Statistics and Graphics Database Workstation is a series of integrated statistics, graphics, and database management programs designed to turn the Lisa into a professional research and management workstation. Users will be able to run multiple copies of *Statpro* in different windows and cut and paste between *Statpro* and other applications such as *LisaWrite*. $1,995. Wadsworth Professional Software, Statler Office Building, 20 Park Plaza, Boston, MA 02116; (800) 322-2208.

Macintosh Goes Adventuring

This month, Infocom is releasing Macintosh versions of its line of text adventures. The programs, translated from the Apple II versions, contain few changes from the originals. Each game accepts compound sentence commands and has a large vocabulary.

The Zork trilogy chronicles the history of a great lost empire that once existed underground. *Zorks I* and *II* use standard scoring and standard goals, while *Zork III* has an unusual scoring system that awards points for appropriate actions, not the solving of puzzles. *Zork I*, $39.95; *Zork II* and *Zork III*, $49.95 each.

Infidel stars an antihero in a desperate quest for treasure in the Middle East. $49.95.

In *Deadline*, the detective has twelve hours to find out who killed an unpopular millionaire. Careless detectives can meet their end, while slow ones may miss important clues. $59.95.

In *Starcross*, the hero is a space prospector confronted with the mystery of an apparently abandoned alien ship. $59.95.

Witness is a campy 1930s murder mystery that reflects the then-popular style of pulp detective fiction. $39.95.

Suspended is a space adventure requiring players to control six robots that can act simultaneously. Each robot has an individual character and special abilities. $59.95.

Enchanter is the first of a trilogy sequel to the Zorks that expands interaction with the other characters, goes above ground, and features the use of logical magic. $39.95.

Planetfall is a science-fiction text adventure in which a lovable robot steals the show. $39.95.

Seastalker casts the player as the main character in a submarine quest to save the world's first undersea research station. The game was designed for beginners and is recommended for those nine years old and up. $39.95.

Sorcerer is the second release in the sequel to the Zork series. Magic is emphasized. *Sorcerer*'s parser understands a vocabulary of more than 1,000 words. $49.95. Infocom, 55 Wheeler Street, Cambridge, MA 02138; (617) 492-1031.

IBM Mainframe Interfaces

AppleLine for the Macintosh and Lisa is a coaxial attachment unit that lets users take information from an IBM or IBM-compatible mainframe and use it at the personal computer level. In most cases, a user can simply unplug the terminal and replace it with an Apple. $1,295.

The Apple Cluster Controller provides an alternate method of micro-to-mainframe communication for businesses that have not installed 3270 cluster controllers and 3278 terminals. With the Apple Cluster Controller, a

Mac or Lisa can be linked directly to the network or connected by modem from remote locations. Terminal software is necessary. Three-port version, $4,500; seven-port version, $7,000. Apple Computer, 20525 Mariani Avenue, Cupertino, CA 95014; (800) 662-9238.

Science-Fiction Adventure

Forbidden Quest is a futuristic text adventure involving journeys on two starships and three planets. The game features mouse control of the most commonly used commands, as well as on-line multilevel hints. Other clues are contained in five prints by legendary comic book artist Wally Wood. $44.95. Pryority Software, Box 221959, Carmel, CA 93922; (408) 625-0125.

Mainframe Graphics Interface

Tekalike allows the Lisa to operate as a graphics terminal with any host computer supporting the Tektronix 401X family of terminals. The program allows the Tektronix commands to be saved and written to disk for off-line processing and saved commands to be redrawn, plotted, and zoomed. $200. Mesa Graphics, Box 506, Los Alamos, NM 87544; (505) 672-1998.

Lisa UniPlus+ Database

Unify is a relational database management system that runs under UniPlus+ and is menu-driven, usable by nontechnical programmers, and features a complete set of development tools, including Sequel 2, C, and an RM/Cobol-language interface. $1,495. Unify Corporation, 9570 Southwest Barbur, Portland, OR 97219; (503) 245-6585.

Xenix Cobol Software

The Santa Cruz Operation is releasing *Micro Focus Level II Cobol*, *High Performance Cobol*, and *Cobol Development Tools*. The first program is a compact, high-level Cobol that contains a dynamic subprogram loading that allows the creation of programs that would not ordinarily fit in the system's memory. $795. *High Performance Cobol* has a native code generator and runs faster than the compact Cobol. $1,595. *Cobol Development Tools* contains a source-level debugger and a data entry program generator (a facility that generates Cobol code for graphics). $795. The Santa Cruz Operation, 500 Chestnut Street, Santa Cruz, CA 95060; (408) 425-7222.

CRTplus

CRTplus is a decision support tool for financial institutions. The program, which runs on both the Lisa and the Macintosh, helps banks, savings-and-loan institutions, and credit unions develop new customers and keep old ones by improving the availability of product information, as well as by aiding the cross-selling of financial products and services. *CRTplus* performs a variety of financial calculations including certificate-of-deposit and early-withdrawal analyses, as well as analysis of IRA accounts, installment loan alternatives, loan amortization, and taxable versus nontaxable investment strategies. *CRTplus* also gives customers personalized printouts with information regarding various calculations and services. Approximately $995; annual maintenance fee, $195. Aurora Systems, 2423 American Lane, Madison, WI 53704; (608) 249-5875.

MacPuzzle

MacPuzzle turns any drawing created on the Macintosh into a jigsaw puzzle that can be reassembled using the mouse. The user gets to specify the number of pieces into which the drawing will be shattered. Approximately $30 to $50. Industrial Computations, 40 Washington Street, Wellesley, MA 02181; (617) 235-5080.

Lisa UniPlus+ Software

UniPress Software has released a complete line of programs that run under their UniPlus+ version of Unix. The *Unix System V* includes a full multiuser (up to eight people) UniPlus+ operating system with C development and text processing tools. $1,495.

UniCalc is a spreadsheet that displays in 255 rows by sixty-four columns and configures the screen so that it can be divided into two windows for easy viewing. *UniCalc* contains many math and logic functions and allows extensive formatting, including exponential and integer presentation, numeric left/right adjustment, floating point, dollars and cents, and text left/right adjustment. $350.

Q-Calc is another spreadsheet with math and logic functions and a model that displays in 1,000 rows by 18,000 columns. *Q-Calc* features sorting and searching and communicates with Unix through pipes and filters. $750.

The *Unify Database* is a fully relational database that is designed to be easily programmable and features a simple query language. $1,495.

UniPress has also released several programming languages for the UniPlus+ system, including SVS Fortran, $600; SVS Pascal, $600; SVS Basic Plus+, $400; RM/Cobol, $1,250; and Irvine ADA, $3,500.

Lex is an interactive word processor with extensive cut-and-paste facilities and a built-in four-function calculator that enables computations to be inserted into documents. Other built-in features include a customer/contact database, a one-hundred-thousand-word dictionary, and a mass-mailing facility. *Lex* is menu-driven, has global search and replace, and formats text by using rulers. $750.

UniPress has also released a line of hard disks that range from 20 to 100 megabytes. Starting price, $3,900.

EMACS is a full-screen multiwindow text editor that contains built-in MLisp files and macros for extensibility. The multiple on-screen windows are user-definable, permitting several files or different portions of the same file to be edited simultaneously and permitting interwindow communication. Binary version, $395; source code version, $995.

UniPress has also released a four-port serial card that allows four terminals to plug into the back of a Lisa instead of the regular two. Approximately $350. UniPress Software, 1164 Raritan Avenue, Highland Park, NJ 08904; (201) 985-8000.

Lisa Business/Industry Software

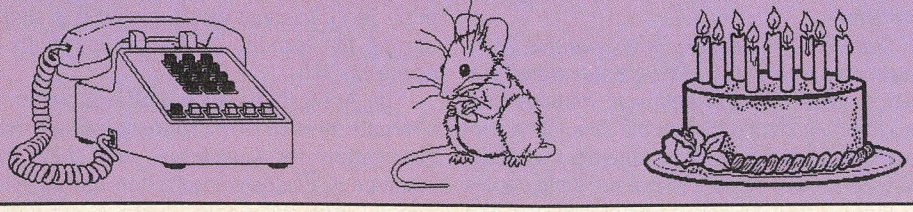
TOM Software has released a complete set of business modules for the Lisa that run under the UniPlus+ operating system. The accounting modules include *Accounts Receivable*, *Accounts Payable*, *Payroll*, and *General Ledger*, while the industry-specific modules include *Public Accountant Business Management*, *Restaurant/Food Service Management Information*, *Property Management Business Information*, and *Not-for-Profit Operations Management*. Other TOM Software modules include *Speed I*, a software applications utility, and *EZ Speed*, a records management program. Each module, $500 to $1,000. TOM Software, 127 S.W. 156th Street, Seattle, WA 98166; (206) 246-7022.

Macintosh Data Manager

Main Street Filer is a filing system and report generator that handles up to 65,000 records in every user-defined file containing up to thirty-six fields of up to forty characters each. The program can maintain four different indexes per file and allows users to retrieve data without time-consuming sorting. The software's print programs include columnar reports,

Macintosh Picture Library

McPic is a push-button library that features 130 pictures for business and home use. The pictures can be used as is or customized with *MacPaint* or *MacWrite*. The artwork can be expanded or reduced, duplicated, or combined with other pictures, and the tones and patterns of each picture can be modified. A picture segment can be selected and used by itself. Words can also be added. $49.95. Magnum Software, 21115 Devonshire Street, Suite 337, Chatsworth, CA 91311; (818) 700-0510.



mailing list reports, labels, envelopes, and Rolodex cards. Any report will be printed exactly as it appears on-screen. Except for data entry, all file design, record retrieval, and reporting functions are mouse-driven. $249. Main Street Software, 1 Harbor Drive, Sausalito, CA 94965; (800) 824-8757.

Lisa Word Processor

R Word has mail merge and math/spreadsheet capabilities. The program does summing per line or math on an individual line, has global search and replace, and can do foreground edits while a file is printing. Printouts look exactly as configured on-screen. *R Word* can also configure for keyboard or RS-232 terminals, and can copy to and from system files. $895. R Systems, 11450 Pagemill Road, Dallas, TX 75243; (214) 343-9188.

Open Systems Accounting Modules

Open Systems has introduced all nine of their Software Fitness Program accounting modules for the Lisa. All the modules require Xenix and run in a multiuser environment. *Accounts Payable* allows users to enter purchases and miscellaneous debits, process requisitions, and print purchase and miscellaneous debit journals. The module also prints checks, cash flow reports, and vendor analysis reports, lists open invoices, and processes prepayments and partial payments.

Accounts Receivable helps users keep track of their company's sales history by producing aging reports, invoices, and statements. Features include the capability to print sales journals and cash receipt journals, list open invoices, and maintain a customer file, an open item file, and a sales history file.

General Ledger summarizes a company's accounting data in any format the user chooses. Features include the capability to enter debit and credit transactions, post recurring entries, print an income and balance sheet, and perform automatic clear, close, and consolidation of accounts.

Inventory has an automatic alert that warns users when and what to reorder. Features include the capability to print status reports, alert reports, evaluation reports, detail lists, sales-analysis reports, and price lists as well as to maintain an inventory file and process quantity adjustments.

Job Cost records costs and revenues on each manufactured job. Features include the capability to maintain a jobs file, perform automatic job completion, process cost adjustments, and print jobs listings, job detail reports, job profitability reports, billing summary reports, and overhead allocation reports.

Payroll allows labor expense distribution and reporting by department and job. The module processes and prints checks and allows employee and department inquiries.

Purchase Order Processing allows users to track orders from the purchase to the receipt of goods, including returned goods and changed orders. This module interfaces with *Accounts Payable* and optionally interfaces with *General Ledger*, *Inventory*, and *Job Cost*.

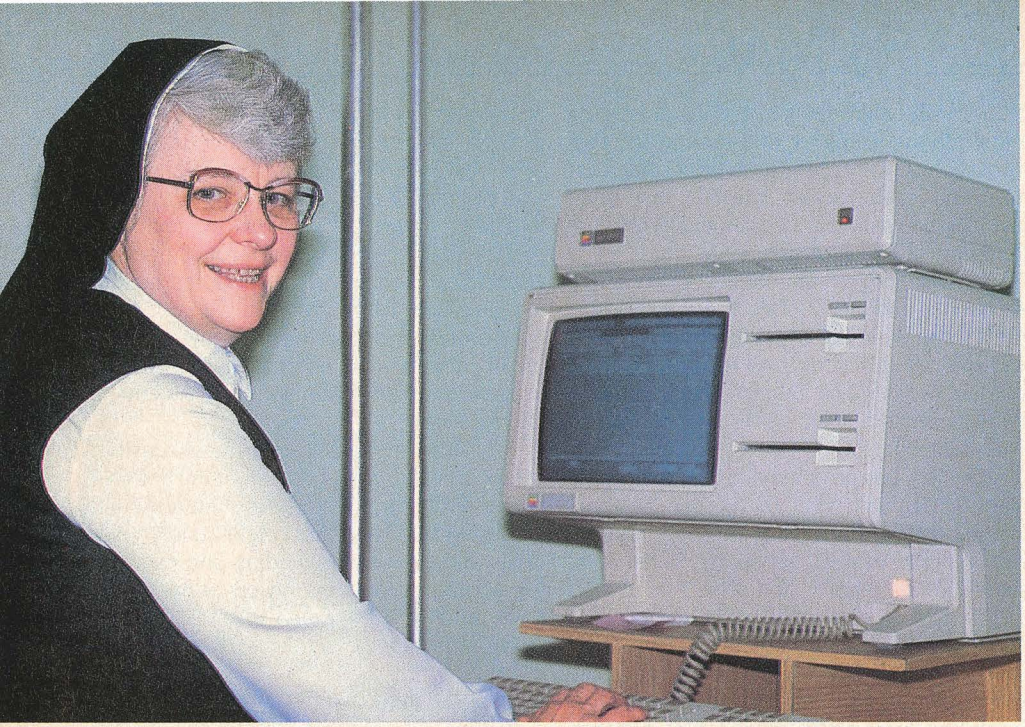
Sales Order Processing automatically processes orders from entry through packing, verification, and invoicing. Features include the capability to handle returned goods and back orders; print packing slips, invoices, and credit memos; and interface with *Accounts Receivable* to produce customer statements and sales analysis reports.

The Team Manager allows users to customize reports and format data from the Software Fitness Program. It includes a predefined data dictionary with more than eight hundred data fields and provides ascending and descending sorting capabilities on up to five fields. Each module, $695. Open Systems, 430 Oak Grove, Minneapolis, MN 55403; (612) 870-3515.

Macintosh Pensate

Pensate is a chesslike thinking game with new tactics. The computer's many pieces move in relation to the one piece controlled by the human player. Each of the ten types of computer pieces has unique rules. $39.95. Penguin Software, Box 311, Geneva, IL 60134; (312) 232-1984.

Please address announcements of new Macintosh and Lisa products and services to MarketWatch, Box 7041, North Hollywood, CA 91605.



A Mouse in the Convent

The Sisters of Christian Charity Has Faith in Lisa Technology

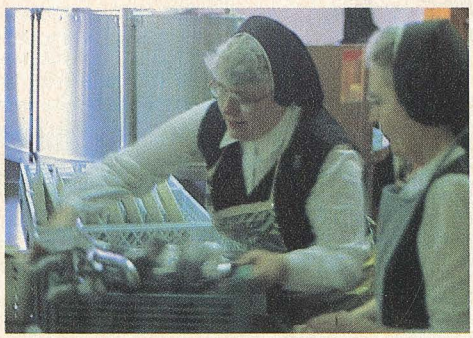
By Mike Ferris

Sister Joachim is just trying to do her job. So far, nothing. All she has is a letter from Apple. Her dealer doesn't know anything; she's already called twice. Apple isn't talking, even if she could afford to call Cupertino, which she can't. So, when is her Lisa going to be upgraded and how long will she have to wait for a 2/10?

Sister Joachim needs to know. After all, her hard disk is almost full and the business functions of a convent of 250 Catholic nuns must go on.

Sometimes, even in a religious community with a mission, time is money. For the Sisters of Christian Charity—a nonprofit organization whose mission is education—financial matters are handled by Sister Mary Grace. Assisting her is Sister Joachim, who uses a Lisa and currently has a few questions for Apple.

Even so, she affectionately refers to the machine as Sister Lisa.



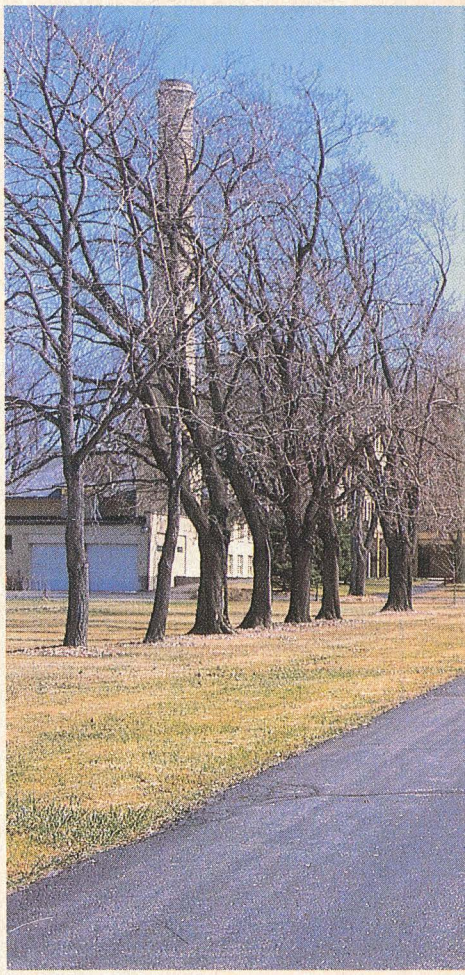
The Sister and the Mouse

A typical day for assistant treasurer Sister Joachim might go something like this. Morning prayer at 6:30. Liturgy at 7:00. Breakfast at 7:30. Work on Lisa starting at 8:00. Lunch at noon. Do the dishes at 12:30, then pray. Continue working on Lisa at 1:30. Vespers in the chapel at 5:00. Dinner at 5:45. Free time for Lisa at 7:00. Watch the news at 10:00, night prayer, then retire.

Sister Joachim—one of the few who had enough faith in Apple technology to purchase the Lisa at $10,000—is a typical Lisa user, enthusiastic about the new machine and eager to explore it. "I'm an Apple fan," says Sister Joachim. "One of the reasons we bought the Lisa when we did was because it was new and going someplace."

The Sisters of Christian Charity, Daughters of the Blessed Virgin Mary of the Immaculate Conception, was founded in Germany in 1849 by Pauline von Mallinckrodt, a German nun. In 1871, when a political Kulturkampf was bent on removing religion from German schools, the Sister was forced to begin searching both North and South America for new foundations for her educational order.

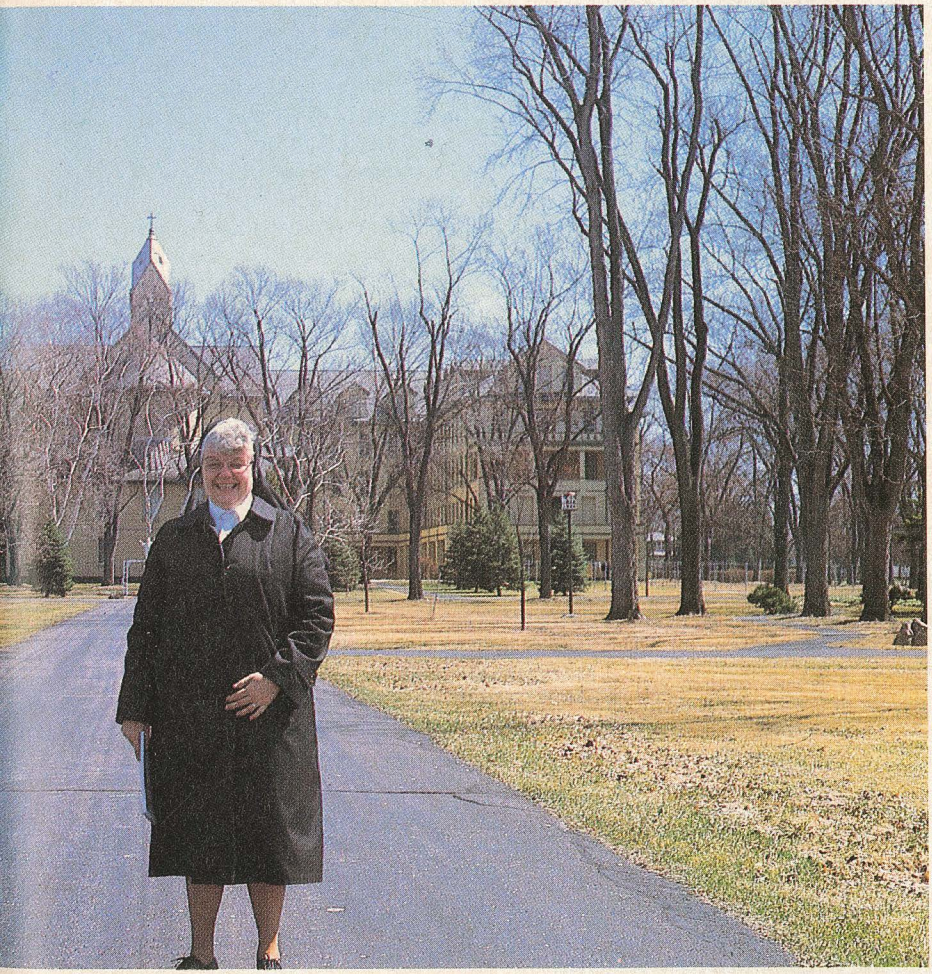
From South America, she traveled up the Mississippi River, establishing educational missions along the way, mostly at the invitation of the local clergy. The 1800s were a time of nationalistic churches in America, and it wasn't uncommon to have a German Catholic church just up the block from a French Catholic or Polish Catholic church. The German Sisters of Christian Charity, brought over in



small groups, taught German Catholic parishioners' children, thus establishing the order in the United States as it exists now.

The Maria Immaculata convent is the mother house of the order's western province. From its location in Wilmette, a suburb of Chicago, nuns take their vows and are sent on missions of education west of the Mississippi, from as far north as Minnesota to as far south as Louisiana. The eastern province, with a mother house of its own, serves the extreme eastern states, from New Jersey to Pennsylvania.

The huge stone and wood building was constructed between 1914 and 1918. "Built like a fort," according to Sister Joachim, it shares two of its four floors with the Mallinckrodt College, where some of the Sisters teach. The first floor, containing a locally renowned law library, and part of the ground floor are devoted to the college. The upper two floors contain the Sisters' rooms; that's also where the Lisa sits in the treasurer's office on the second floor. A surge suppressor protects it from the new freight elevator around the corner, which pulls power every time it's started up.



Sister Joachim gives a tour of the Maria Immaculata grounds.

Sister Joachim entered the mother house as a novice in 1960. Originally from Detroit, where she attended an all-girls high school run by the Sisters, she took her final vows in 1969. She has been sent to teach at sister missions in New Orleans, Chicago, and Prospect Heights, a suburb of Chicago. Four years ago she returned to the mother house to help Sister Mary Grace keep the books.

As an educational order, the Sisters of Christian Charity aren't cloistered behind brick and stained glass. They're educated, informed, and aware; each has specialized teaching skills and some also have specific business responsibilities—for instance, Sister Mary Grace knows bookkeeping, and Sister Joachim now knows computers.

"Sister Mary Grace knows all about the computer revolution," says Sister Joachim of her colleague, who has set up a few files herself on the Lisa.

Lisa Takes the Vows

The electronic revolution at the mother house began with John Voosen, current architect for the convent and a friend of Sister Joachim's. While constructing a physical therapy room for the infirmary, Voosen "was always talking computers," says Sister Joachim. His architectural firm uses several Apples that "do everything but talk back," says the Sister. "He does his billing and does architectural drawings on them."

It was Voosen who set up a weekly show-and-tell session for Sister Joachim on his home Apple so she could see what a computer could do for the convent, which was already pressed with paperwork. "We didn't know computers from anything at the time," says Sister Joachim, "but we knew we eventually had to get one."

On and off for about four months, Voosen demonstrated the machine's capabilities. "One night, for kicks, he showed us how to log on to CompuServe," Sister Joachim says. "We actually saw a flight to Germany that some of our Sisters were taking confirmed."

That was a thrill, she admits, but Sister Joachim had a few questions for Voosen. "Okay, I have this file or thing," the computer novice said. "Can you put it up and have it be of service to me?" As an example for the Sisters, Voosen chose to create a personnel file card on a database. "We worked it out and I saw it could be done," she says. Her next question about computers was, "How much does it cost to get one?"

When it turned out that a personal computer was indeed within their budget, Sister Joachim said to her boss, "Let's go look."

The two Sisters went shopping for a computer last July, accompanied by Voosen. It was he who suggested they try a computer store fifteen miles out of town, one that he knew gave hands-on demonstrations.

At the computer store, Sister Joachim was put through a demo on an IBM PC. "But I had my eye on the Lisa in the next room," she says. So did Sister Mary Grace, it turned out. Both had read newspaper reports of the new machine and seen the commercial for it. "Although that's the dumbest ad, with that guy sitting in the middle of a room. It tells you nothing," says Sister Joachim.

Nonetheless, the Lisa seemed to have everything they were looking for. The technology wasn't going to become outdated, there was database software coming, the screen was easy to understand, and "we could be doing some of our things on it in a few hours," says Sister Joachim. The original $9,995 price tag only made its possible purchase a more agonizing decision for a small operation like the convent.

"We're a small business, only with different needs," Sister Joachim explains. "We think we're big, but we're peanuts in comparison to other companies." Still, the need for a computer like the Lisa was there.

Time was also a factor. Summer retreat was coming up and shopping for a computer would become more and more of a luxury as the business of the convent went on. The retreat would keep the Sisters at the mother house busy with Mass, meditation, and conferences. "We had to take the plunge on the Lisa," says Sister Joachim. "There was no time to wait."

So Sisters Joachim and Mary Grace decided to juggle the budget to make room for the more expensive machine. "We discussed it at the computer store that morning and had decided by noon. After lunch that afternoon, we loaded the Lisa into the car and brought it home.

"Once Sister Mary Grace had said okay to the Lisa, I had no doubt," says Sister Joachim. "I was very, very eager to get started working on it."

Uncovering the Beans of Lisa

Until the computer was moved into their second-floor office, Sister Joachim had it set up on a table in her room. "I played around a little with *LisaProject*. Just kidding, I planned a three-month vacation to Europe for myself," she says. In real life, the Sisters are allowed three weeks off a year, two for visiting family and one for visiting another convent or school.

"I was fascinated by actually having the Lisa here to play with," she says of those first experimental days. "I didn't know beans about the Lisa. I just plowed through the thing and did the samples. Then all kinds of new fields began to open up. As John Voosen would say, 'It blows your mind.' I had listened to him about computers but I didn't believe it. Now I can see all kinds of possibilities."

Now she has visions of a network of Sisters of Christian Charity missions, stretching from Minnesota to New Orleans, all connected by computer. "Is that too big of a dream?" she asks. Maybe not. An even bigger dream is being able to stop using the mainframe general accounting system in Saint Louis that the order subscribes to. By January, the Sister had made a pad of sheets on *LisaCalc* with room for the changing numbers. She now sends this to Saint Louis every month instead of the old form.

In addition to being the financial boss of the convent, Sister Mary Grace, right, was one of Sister Joachim's instructors in high school.

"It's a beginning," she says, "and a lot less of a headache. I hated having to spend all that time and energy lining up numbers on the old form. It was such a mess. Now it's a snap."

One early glitch, involving the printer, wasn't easily resolved. "I could do letters, drawings—anything to mess around—and then print them. But I couldn't get it to print 14 1/2 by 11," says the Sister. "It was a hassle lining up the pages by hand, so I packed up the Lisa and took it back to the computer shop."

It seems that a bar and two screws left in from shipping the machine were keeping the printing head from going all the way to the right. "An I and an E were kind of close together at the end of the line when the printer stopped. That's what clued us in." And this simple discovery was made only after the guts of the machine had been replaced and several knowledgeable computer people had fussed over it. "We still laugh about it," she says.

So far, Sister Joachim has transferred batches of former paperwork to the Lisa. Personnel and health records are now kept on *LisaList*, where the education degrees and schedules of the Sisters are listed, along with the model year and plate numbers of the seven convent cars that are kept there for insurance purposes. "Eventually I want to list repairs on *LisaList*, and bank reconciliation on both *List* and *Calc*," she says.

The convent's general ledger is also kept on *List*. "I've plowed through BPI's *General Ledger*, but so far it's hit and miss," she says. "Their system of numbering is different from *List*'s, and switching over won't be easy."

The chart of bookkeeping accounts is kept on *LisaList* for searching and *LisaCalc* for calculations. The budget is figured using *Calc* and the wages of three part-time employees are too. All these files are kept on both programs because *List*, of course, doesn't calculate. So Sister Joachim often goes through the information with calculator in hand, then transfers the figures to *Calc*. "It's a hassle, but it's easier than it was; I really don't mind.

"The real problem right now is trying to double-space on *LisaList* and have the spaces survive when you go to sort," she says. She's tried numbering the spaces, but they all print out in one blank block at the beginning of the page.

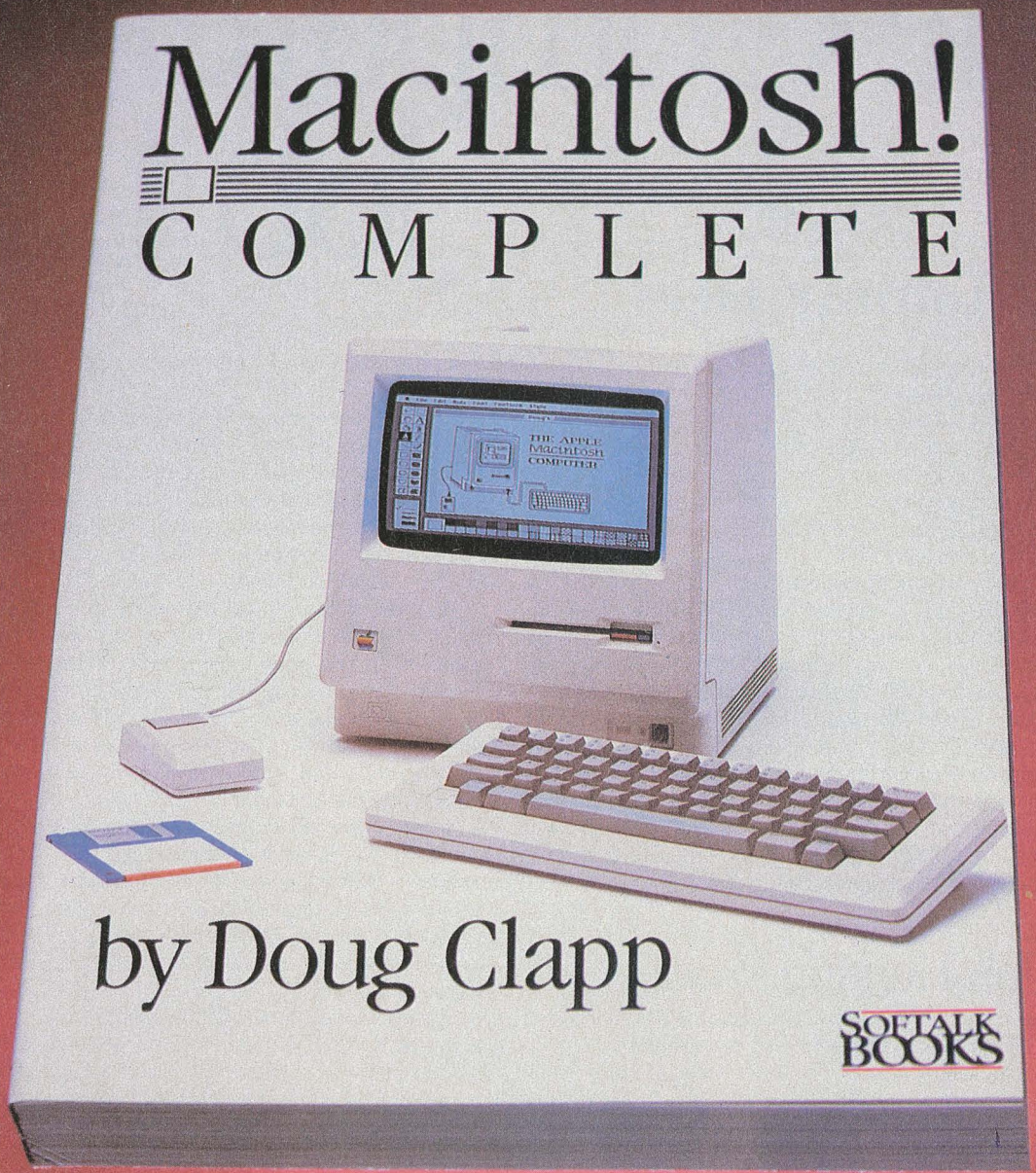
Subscriptions to *Icon* and *Signal*, two newsletters for the Lisa, keep the Sister up to date, giving her both hints and news. Right now, two letters from Apple and BPI notifying her of system changes have got her hanging.

"Should I wait until the 3 1/2-inch upgrade arrives, or should I keep working on the 5 1/4-inch disks?" she asks matter-of-factly. Sister Joachim has only 174 blocks out of 9,960 left on her hard disk. "I'm so eager to get everything switched over that it's frustrating," she says.

Sister Joachim is among the thousands of early Lisa buyers who feel that their commitment to the Lisa should not go unrewarded by Apple. So she waits with all the rest for news of her 2/10 upgrade, but not patiently. "I'm a little upset," she says. "I hope they give us a break on the price [and not charge the full $2,495 Apple is asking for the trade-up]. After all, we were the forerunners."

Her allegiance to Apple seems strong, despite the current slowdown of her computing activities until the question is resolved. The Sister still plans to keep supporting the company's 68000 series. "I would eventually like to see a Macintosh on Sister Mary Grace's desk so she could be tied in to the Lisa," says Sister Joachim. "I think she's ready for one."

Sister Mary Grace would agree.



THE FIRST AND DEFINITIVE BOOK ON THE APPLE MACINTOSH COMPUTER

Written by one of computerdom's most literate authors.

Available now. $19.95.
Order from Softalk Books: 818-980-5074
Visa and MasterCard welcome

$1.50 shipping charge per book.
California residents add 6.5 percent sales tax.
Apple and Macintosh are trademarks of Apple Computer.

The Computerized Job Hunt

By Tommy Gear

When speculating about all the neat applications people will soon be discovering for their Macs, one very practical question comes to mind: How can the Mac be used in finding a job? For help in answering this question we contacted Richard N. Bolles, author of the bestselling book, *What Color Is Your Parachute?*

Parachute serves as a practical how-to manual, not only for the job hunter but for anyone interested in learning more about who they are and what they can become.

An authority on the job-hunting process, Bolles directs the National Career Development Project, a nonprofit organization that presents workshops and publishes a newsletter for job hunters and career-changers nationwide. He also is working closely with Brode. und Software on a disk version of *Parachute*'s Quick Job-Hunting Map. This tool helps you identify your skills, determine where you most want to use them, and succeed in finding the job you want.

Bolles has strong convictions on the dos and don'ts of job hunting with a computer—and the Mac in particular. In a recent interview with *ST.Mac*, he discussed the job-hunting process, the uses and abuses of résumés, the validity of occupational forecasting, and the wisdom of seeking employment in a high-tech field.

On the Job-Hunting Process

Let's look at the computer in terms of the tasks a job hunter needs to do in order to be successful and ask, "Can the computer help?"

The most obvious task you as a job hunter need to do is to sit down and organize information about yourself. I suggest you write down seven episodes from your life that you take some pride in and in which you set out to achieve something and succeeded in achieving it. Even if you were working with others, these should be areas in which you feel you made a significant contribution so you can truly claim that achievement to be yours. If you don't want to write seven achievements, I suggest that you write an autobiography.

Any computer with the simplest word processor would be very serviceable for doing these things. It would make the whole process fun, and most people can write faster on a computer than they can write on a typewriter. The computer offers an ideal medium for getting these thoughts out on the screen, then onto paper.

No job hunter should ever go out looking for a job without some idea of what it is they do well, or what it is about what they do that distinguishes them from other people. You need to be able to respond to a boss who's saying, "I have fifteen people applying for this job—what distinguishes you from them?" And it can't just be in the nature of the skills; it's got to be in the way you perform those skills. What the computer should provide is a list of possible skills.

The second task for you, the job hunter, is to begin keeping notes on the places you go to visit in your job hunting. Record who it is you should call back and so forth. All job hunting is nothing but an information search. Those naive persons who first go out on a job hunt think they know what the information search is. They think it's the search for the availability of a job.

But you can't just look for a vacancy, because many organizations are in the process of creating new positions. If you're walking in at that point, rather than when a vacancy has already been announced, you have a much better chance of getting the job. And there's often quite a gap between the time an organization decides it has to hire somebody and the time it actually moves to do it.

For the job hunt to be really effective, the information search must equally be inside the person. The more effective job hunts are always those done by people who have looked for information about themselves as much as they've looked for information about what's available in the world of work.

Suppose, like a typical job hunter, you go to a number of places looking for jobs. Suddenly you wake up one morning and say, "I know what's been bothering me about all these places I've been visiting. They all have over a hundred employees and I realize I'd rather work in a smaller place that employs only about twenty-five people. I remember visiting a place three weeks ago that was small like that, and I liked it, but I didn't realize why at the time. If only I could remember which place that was."

I urge every job hunter: Keep some kind of files on the places you've visited because you just can't trust your memory. Database programs that allow you to search under any word could be very useful for managing this kind of information, but there's a deficiency in those databases that forces you to define key words. Suppose there's a category you haven't thought of because it wasn't important to you at the time.

It's a sign of the times that the creators of these database programs are realizing that sometimes you need to go back in search of information that was insignificant to you when you first entered it, information that only later has become very important.

On Résumés

A person who was untutored about the nature of the job hunt might look at a computer and say, "Aha! The greatest thing the computer can do for the job hunter is to help develop a résumé." This is the area I dispute the most.

A résumé is actually four different documents, combined into one page. First, a résumé is a way of organizing information for job hunters so they can see their own work history more clearly. You would not need a résumé for this purpose if you had worked on your list of seven achievements mentioned earlier.

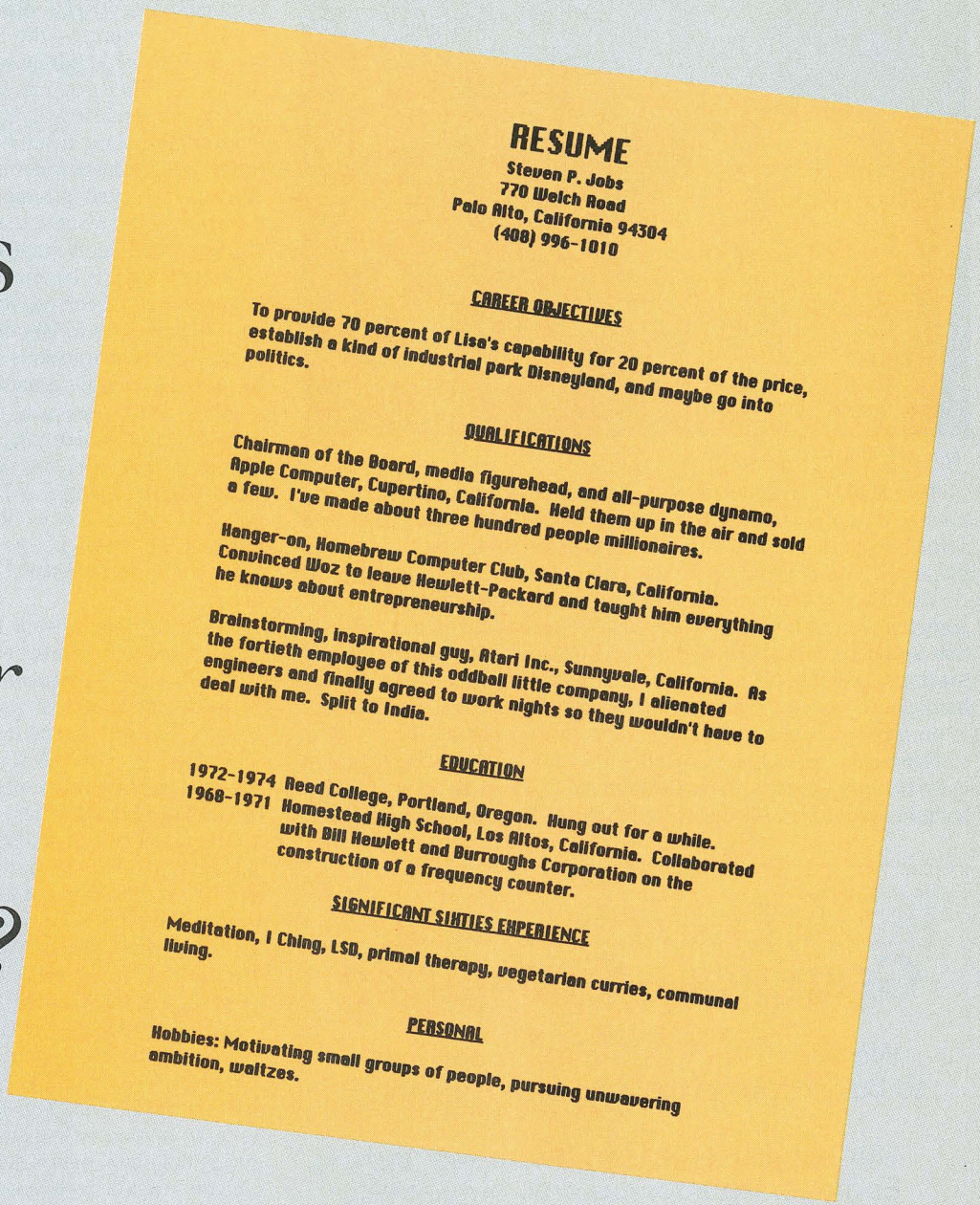
Second, a résumé is used as an extended calling card to get your foot in the door. You slip it into the mail hoping that the people receiving it will open it up, read it, and conclude from it that you are just the kind of person they would like to talk to.

I guarantee you that a résumé sent through the mail and printed on a dot-matrix printer is going to get short shrift in today's world from many, if not most, employers.

I receive résumés all the time from people who ask me to look them over before they send them out.

Third, if the employer who receives your résumé decides that you're interesting enough to interview, he or she is probably going to use your résumé as the agenda for the interview. Many employers are not skilled in conducting interviews and will use your résumé as an agenda

Some Thoughts from the Author of *What Color is Your Parachute?*



RESUME

Steven P. Jobs
770 Welch Road
Palo Alto, California 94304
(408) 996-1010

CAREER OBJECTIVES

To provide 70 percent of Lisa's capability for 20 percent of the price, establish a kind of industrial park Disneyland, and maybe go into politics.

QUALIFICATIONS

Chairman of the Board, media figurehead, and all-purpose dynamo, Apple Computer, Cupertino, California. Held them up in the air and sold a few. I've made about three hundred people millionaires.

Hanger-on, Homebrew Computer Club, Santa Clara, California. Convinced Woz to leave Hewlett-Packard and taught him everything he knows about entrepreneurship.

Brainstorming, inspirational guy, Atari Inc., Sunnyvale, California. As the fortieth employee of this oddball little company, I alienated engineers and finally agreed to work nights so they wouldn't have to deal with me. Split to India.

EDUCATION

1972-1974 Reed College, Portland, Oregon. Hung out for a while.
1968-1971 Homestead High School, Los Altos, California. Collaborated with Bill Hewlett and Burroughs Corporation on the construction of a frequency counter.

SIGNIFICANT SIXTIES EXPERIENCE

Meditation, I Ching, LSD, primal therapy, vegetarian curries, communal living.

PERSONAL

Hobbies: Motivating small groups of people, pursuing unwavering ambition, waltzes.

by default. Employers often fall back on the résumé in order to hide their ignorance on how to properly conduct a job interview. If the employer is a bad interviewer, he will complete the interview and realize only later that he didn't ask you what he really needed to know in order to make a decision. As a job hunter you've got to make sure that during the interview the employer is getting all the information he needs—even if he isn't organized enough to get it all himself.

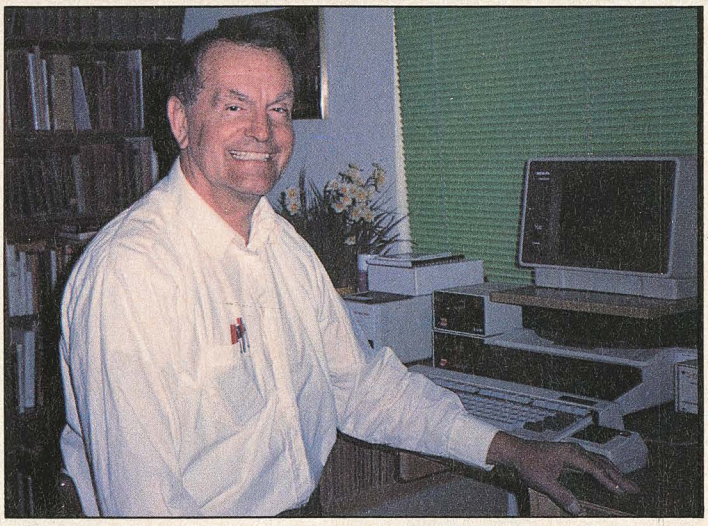
In the new edition of *Parachute* there's a list of questions from the employers' point of view, and in the book I explore why they ask these particular questions. Always the answer is fear. Every question the employer will ask you is out of fear. What is the fear? Well, it's either fear of the past, the present, or the future. If the employer asks why you quit your last job, what he's really asking is, "If I hire you how do I know you're not going to quit me?" But if the employer asks, "If I give you this job, where do you see yourself in five years?" you can begin to see a time sequence. The questions are moving from the past to the present into the future.

Listen to the time sequence of the questions being asked in an interview. If the questions are all in the past, that's not so favorable to you. If the questions start to move into the present, this is more favorable, and if the questions are moving into the future, this is most favorable. It indicates the employer is thinking of hiring you.

The résumé as agenda is not where the job hunter's best interests lie. A résumé is, by definition, an obituary of your past; it's a statement of what you have done. But in an interview you want to be future-directed.

There's one use left, and that is as something to leave behind with the employer as a reminder. I personally believe that if you're going to try to use the résumé as an extended calling card, or by default as the agenda for an interview, you had better get it typed by a professional service on a nice typewriter if you can't do it yourself.

The National Career Development Project is the one place in the country where people can write and get answers to any question they have about job hunting without charge. I find that I rapidly lose interest in résumés printed on a dot-matrix printer. First of all, it tells me a couple of things about the person sending the résumé: It tells me that the person didn't care enough to take the résumé down to a local print shop and have it printed decently. This person didn't even bother to go to a secretarial service and have it typed up nicely. The first impression I get from this is that the person is trying to save time and cut corners. But if that's how he conducts his job hunt, what is he going to do when I hire him?



Job-hunting authority Richard N. Bolles.

The other thing is that I happen to be fifty-seven years old and my eyesight is not as terrific as it used to be. It's often difficult to read something printed on a dot-matrix because it's always several degrees lighter than other things we're used to reading.

Third, some dot-matrix printers are better than others, obviously. Sometimes the letters are so poorly formed, and the dots so visible, that a reader could get carried away by what his eyes have to do with the dots. An employer might start realizing that he has to work awfully hard just to read the résumé. The job hunter is asking the employer to do more work than he might personally be willing to do.

Now when we start to talk about the résumé as something you leave behind you *after* you've been interviewed, we're into a different ballpark altogether, because the employer has now met you. In a résumé that's sent to the employer after the interview, the Macintosh would offer the possibility of doing some really creative headings, printing different sections in different fonts, and so forth.

After—or during—the interview, tell the employer honestly that you don't have a résumé with you but offer to drop one in the mail right away. Then you can go home and, with a machine as creative and ground-breaking as the Macintosh, be able to develop a really interesting-looking résumé. It will generally be forgiven that it's not typed on a good typewriter because the employer has seen you already. The résumé is just to remind him of who you are, and, in fact, some creativity in the résumé may be much more permissible because he can interpret that in the light of who he knows you to be.

If the person is a very logical, analytical, straight-laced, cerebral, left-brain type of person, this might not go over too big. But if you're applying for a job where any kind of creativity is going to be desired, and where the employer is something of a creative individual himself, he probably would appreciate it.

There are two caveats: Even the Macintosh, good as it is, is not the vehicle on which you ought to be hammering out a résumé if, after your interview, you realize you're dealing with a very straight-laced type of employer. Whether this is generally changing out there in the whole world of work is of no interest to you if the guy that has interviewed you is the one holdout. Trends are not useful to know about in this case. What you need are strategies for dealing with real people.

So many people want to hear what the trends are. You will be dealing only with one individual at a time. If every other employer in the world accepts a Macintosh-hammered-out résumé and the employer in the organization that you want a job from doesn't, the knowledge of that trend will do you no good whatsoever.

The other caveat: A résumé done on a Macintosh is clearly not acceptable when you're dealing with a committee, when one person has met you but the others have not. Here you're undoubtedly one of a number of people competing for that job. The person who has actually done the first round of interviewing will toss onto the table all the résumés from the different candidates.

From one point of view, your Macintosh résumé will stand out from the others because it looks inventive and interesting. If you're dealing with an art agency or an organization where creativity is to be rewarded, that's a good résumé to have. But if the members of the hiring committee who have not met you are now going to have their first impressions of you through this résumé—and yours looks like a kid's résumé while the others look like they've been produced by adults familiar with the language of the workaday world—they're going to immediately dismiss your Mac résumé. You should inquire during the initial interview if the hiring decision will be made by the individual who's interviewing you or by a committee. They will usually tell you that anyway.

On Computer-Based Job-Hunting Systems

There are a number of computerized vocational programs that have been around for years and a number of career counselors who are ecstatic about these programs. They attempt to do what is essentially a matching of skills and personal inclinations to specific job categories. There are several types of problems with programs of this kind.

Such programs usually operate out of a very limited database that contains only a select number of occupations. I don't see any reason why it wouldn't be possible to utilize a huge database, so this might be only a temporary limitation.

Second, the basic thrust of all vocational programs that presently exist on computers is that they are matching systems that tend to narrow down the base of choices more and more. In other words, the more questions they ask the person, the more they narrow down possible employment alternatives for that person. In this way they attempt to find that one job that's right for you—and sometimes they

If you had asked me three years ago, "Don't you think the computer would be a wonderful tool to empower the job hunter?" I would have said, "It stinks."

narrow things down so much that they conclude there are no jobs right for you.

A matching program on a computer works by excluding things. When you answer a question, the computer uses this response to eliminate certain areas that it determines don't apply. The matching concept is so easy to do in concept and so difficult to do in practice, because the criteria used are not mutually exclusive. If you look at this process logically you can see that the computer eliminates a number of job alternatives with the answer to one question, but some of those jobs may need to be brought back the minute an evaluation takes place based on another criterion. Since the computer operates in sequential fashion, it's already locked those other jobs out.

On Computers and a Sense of Empowerment

When you're faced with a crisis, like needing to find a job, you have two choices. The first is to find somebody who knows how to bail you out of the crisis on the assumption that that's the only time you're going to be in that crisis. That's the kind of solution we opt for when we have a broken ankle and we go to a doctor. We don't think we're likely to be in that crisis again, and the doctor knows far more than we do about how to cure it, so we put ourselves in his hands.

There is an attitude about job hunting that says the same thing: Job hunting is a crisis, and what we need are good job counselors, or good employment agencies, and they will rescue us. This takes the point of view that job hunting is a sort of nonrepeating crisis. But job hunting and career change are not really a crisis in our culture; they are a recurrent phenomenon.

The second choice is to take the crisis that you are facing and use it as an occasion to learn how to handle it yourself. Thereafter, when you face that same kind of crisis, again you will know how to deal with it rather than having to always find someone to rescue you.

The role of the computer in that kind of understanding is very clearly a wonderful potential that has not yet been realized.

If you had asked me three years ago, "Don't you think the computer would be a wonderful tool to help empower the job hunter?" I would have said, "It stinks." It would have taken a whole set of special skills then just to operate the computer, skills that may or may not be parallel to the skills a person wants to use.

Macintosh, windows, the mouse, integrated software—all these developments represent a decrease in the number of skills an individual has to have just to operate the computer. That's what gives it more and more potential as an empowering tool. Some people find those skills congenial, and some people find them very alien. For those who find these skills alien, the computer is not a good tool.

On Occupational Forecasting and High-Tech Jobs

The problems with occupational forecasting are several. A researcher might interview a number of employers, and let's suppose these are high-tech employers who know they are going to need, say, four hundred twenty-five thousand people over the next five years. To be really safe the employers will tell the researcher they need five hundred thousand to six hundred thousand people, to ensure that they'll be protected in case they made an inaccurate guess as to how many they really need. The researcher has taken a number of months to gather the data, and during that time, of course, a number of people have been hired for those four hundred twenty-five thousand jobs. When the researcher decides to publish the occupational forecast it may take another year for the report or book to wind its way through the presses. During that year still more people will have been hired.

The study might acknowledge that high-tech jobs basically require at least two years of good college or technical school training—maybe four years and sometimes six for a master's degree. In that length of time, from when the report was first published, a lot more people will have been hired for those jobs. The upshot of the matter is that it's very possible by the time a person reads such a forecast, and then acquires the necessary training for these jobs, and then goes out to look, that the jobs won't be there anymore.

With some amusement I've said that if someone wants to take occupational forecasting seriously, what they really ought to do is look at the jobs that are absolutely filled, that nobody wants, and go for those. By the time the person is eligible for those jobs, the pendulum will have swung back in their favor. So the minute I hear people saying that high-tech fields are where all the jobs are going to be, I have serious questions. There are 104 million people in the labor force right now, and you can't tell them all to go into high-tech.

The real issue for the individual is, if you know what your skills are, and particularly what your favorite skills are, could you find a job doing that which you most enjoy doing? The answer is yes.

MISCELLANEA

Club Mac

It's not a popular health spa but a national user group for a popular computer, the Macintosh. According to founder Stephan K. Elliott, Club Mac's emphasis is on pooling information for everyone's benefit. To allow an exchange of ideas across the country, Club

Mac plans to provide members with a Macintosh communications program, a monthly newsletter, access to an on-line idea exchange, and a professionally staffed help line. The on-line service will list available Macintosh programs and accessories, as well as the interests and backgrounds of other Club Mac members. The bulletin board system will also enable Club Mac members to share programs, find solutions, and even fill community service needs. The club's founders expect people to join at the rate of one thousand per month. "It's exciting to be at the center of a whole new communications medium. We want to help our members break new ground every day," says Elliott. To join Club Mac, send $35 for annual dues to Club Mac, 735 Walnut Street, Boulder, CO 80302, or call (303) 449-5533.

MacWrite Update Update

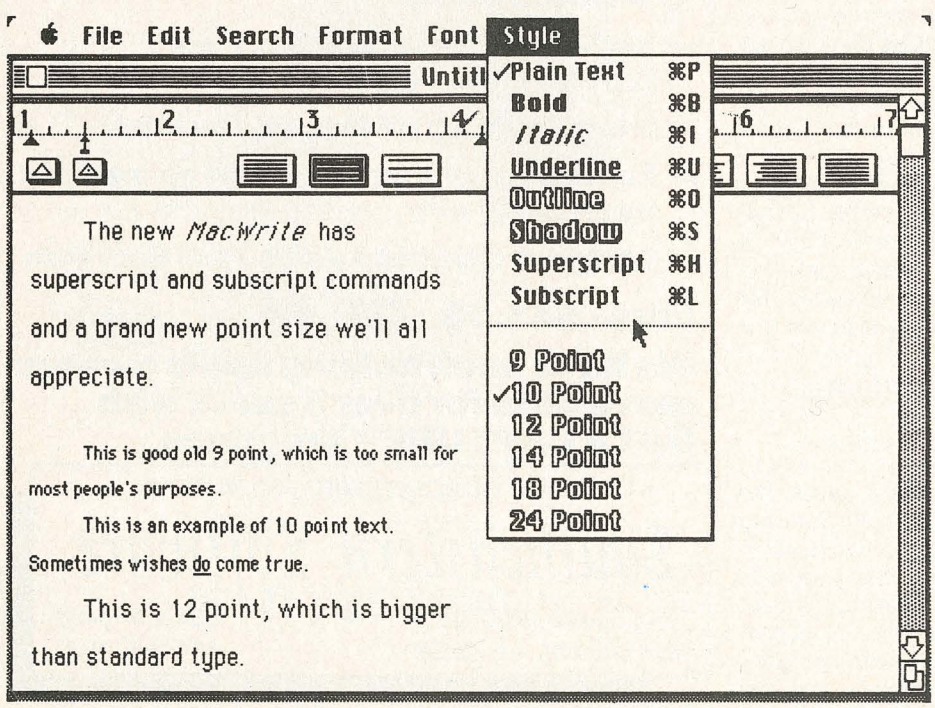
The new version of *MacWrite* just goes to show how meaningful the little things in life can be. Little things like a new type size: The much desired ten-point now exists, at least in Geneva (the font, not the city). Also making a well-timed debut are superscripted and subscripted type styles.

The new *MacWrite* is accompanied by an updated *MacPaint* and a new version of the Finder. The changes to *MacPaint* are still a little sketchy (details next issue) but are said to include a higher quality print mode and ten-point type.

The new Finder will be a relief to single-disk-drive users (although the greatest relief would still be a second disk drive), as it does file copies in fewer disk swaps. There are also a couple of changes to the desktop system. One is a new option on the Special menu called Set Startup. This allows you to specify that a particular application is to start up immediately when you boot the disk. The other change is that you don't need to select the disk icon before using the Eject option.

We'll have to wait and see what happens if you're using two disk drives, but if Apple wants this feature to be really intuitive, they should make it eject the disk that holds whatever file is selected at the time.

These new versions of the software will be included with Macintosh beginning any day now. Owners of older versions will be able to get a free update from their dealers—all they have to do is bring in an old *Write/Paint* disk and a System Disk and the dealer will exchange them for the new ones.



Apples Mean Business

Sears Roebuck and Company and Businessland, two retail chains that previously declined to sell Apple computers, have announced that they will carry Macintosh and Lisa. The decisions are seen by many as an important validation of Apple's strategy to market Macintosh and Lisa as business machines despite skepticism from those who said Apple can be successful only with computers targeted toward the home.

Sears will sell the Macintosh and the three models of the Lisa 2 in all sixty Sears Business Systems Centers beginning in June. The Chicago-based retailer plans to open forty more business centers this year and another fifty in 1985. The centers sell electronic equipment for businesses and institutions. Sears will not carry the IIe or the IIc, which are still considered mainly home computers.

Businessland, perhaps an even more significant marketplace for Apple because it caters solely to Fortune 1000 companies, will now sell Macintosh and Lisa alongside the IBM PC, Eagle, Compaq, and the Burroughs

B20 line of computers in its twenty-five stores throughout the West and Southwest. This is the first time Businessland has considered any Apple computer appropriate to the needs of its competitive white-collar clientele.

Alan Kay Switches Tracks

Alan Kay: inventor of windows, founder of Xerox Palo Alto Research Center, and creative spark behind the Dynabook, the concept of a supercomputer in a notebook-sized package.

Alan Kay: an extremely rich man, thanks to Atari, the company that paid Kay a rumored $5 million to leave Xerox and join the folks that gave us *Pac-Man*.

Alan Kay: subject of a *ST.Mac* interview in our April issue. Past readers may still remember Kay's comments about Macintosh, which included: "Macintosh is an implementation of the designs of 1971. Big deal!" and "Personal computers are the *Laverne and Shirley* of computing."

Now, get ready for...Alan Kay: Apple's newest employee.

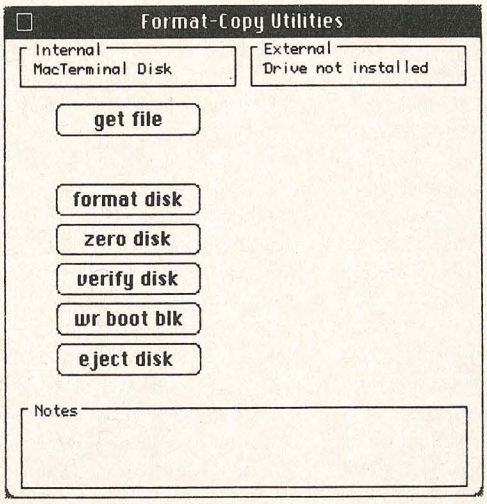
No kidding. Apple has just confirmed that Kay will become a member of Apple's prestigious "Apple Fellows." The few Apple Fellows (four at last count, pre-Kay) include Bill Atkinson, Steve Wozniak, Rod Holt, who designed the power supply for the Apple II, and Richard Page, who spearheaded the team that picked the 68000 microprocessor (and designed the Lisa software development tools).

The Fellows are the technical equivalents of "vice presidents" in Apple's management team. Apple Fellows are given sabbaticals (one year to do research, renewable), hefty research budgets, and hefty bonuses, stock options, and other corporate goodies.

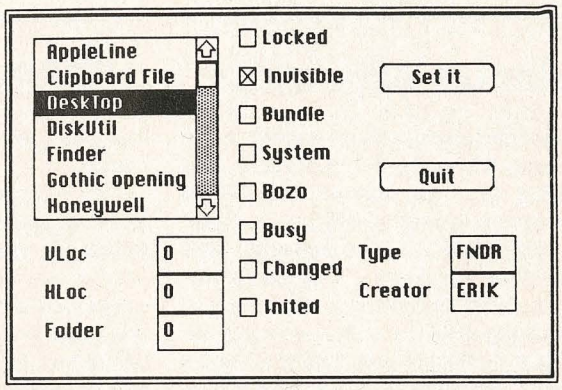
What will Kay do at Apple? Nobody's telling. No one, in fact, is sure what he did at Atari, since nothing's come from Atari that even faintly smacks of Kay's genius.

What did Apple pay Kay to make the move? Beats us. But an off-the-cuff guess might be: a lot. Or more.

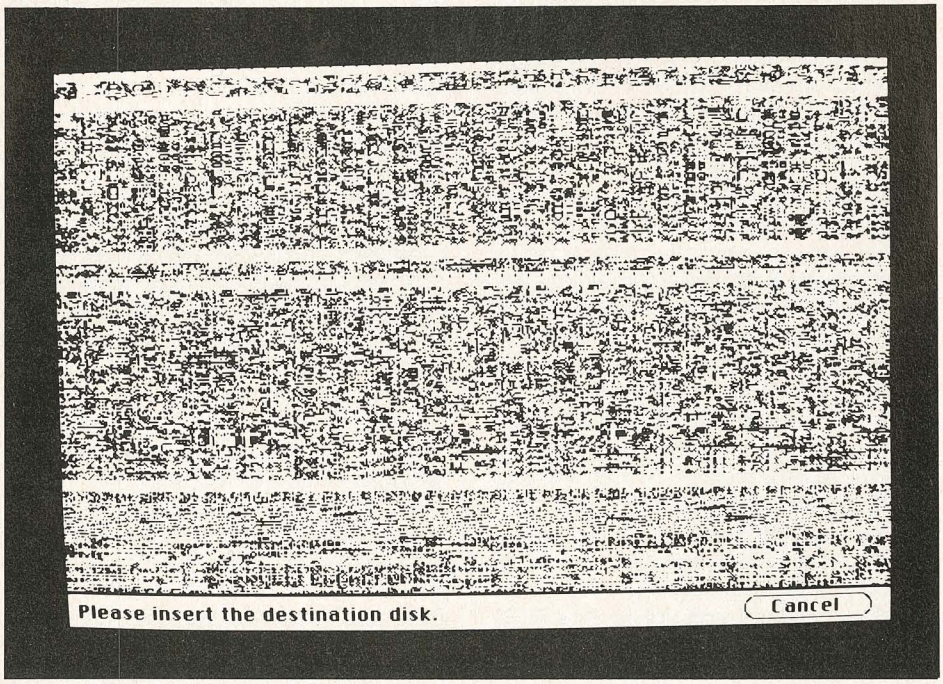
In other news, Domingo Valdez will play out his option with Microsoft, Paul Schwartz is considering going with IBM for an estimated $1 million in a three-year package, and Freddy Landsbury is reportedly being traded from Commodore to Kaypro for $1.7 million, three assembly language programmers, and a left-handed relief chip designer.



DiskUtil



SetFile



DiskCopy

The Techie Underside of Macintosh

Floating around developerland are some strange and wonderful (and some not so wonderful) Mac utilities. Some of these programs will probably never see the light of a commercial disk label, but they're kind of fun—if you're starved for weird functions and confusing jargon, that is.

The first of these beauties is SetFile. This program can be used to change a file's type to load it into an application other than the one that created it. For instance, the Type parameter of a text file created with Microsoft Basic can be altered so that *MacWrite* will be able to load the file. The utility also allows you to see (at least the names of) invisible files, such as DeskTop.

There are other parameters that SetFile allows you to change. Vloc and Hloc appear to control the icon's location within the disk window, and Folder specifies which folder a file resides in. Most of the other parameters are clear as mud, especially Bozo.

Your life will be much easier if you never get your hands on SetFile, which is probably meant only for developers. It seems to be prone to crashing disks when used, shall we say, experimentally.

The next program, DiskUtil, is interest-

ing, but it's hard to say for what purpose. Before the Mac (and the final system software) was released, early versions of the Finder had some difficulty making bootable copies of disks; DiskUtil's "wr boot blk" (or "write boot block," in English—we think) function provided that crucial capability. This utility also will probably not see commercial distribution. At least not until someone can demonstrate that it does anything.

The last utility, DiskCopy, is the sleeper of the lot. This one's being distributed on the new *Write/Paint* and System disks, which is great news because it holds the solution to the major problem in copying disks with the Finder—that problem being death by old age of the person swapping disks. DiskCopy takes over the entire system, throws out most of the stuff in memory, uses all but the bottom half inch of the screen as part of its copy buffer, and copies a whole disk in four swaps. It's fun to watch, too. Raw data displayed on a graphics screen makes such interesting patterns.

Real Writers Don't Use....

This may be our last "while Mac was being created" story. There are only so many of these things, you know.

Okay. Mac is being created. Hardware people are hardwiring, software people are coding, and, of course, the writers are writing: owner's guides, *MacWrite* manuals, *MacPaint* manuals, and pounds and pounds of internal documentation that finally became *Inside Macintosh*, a complete set of everything-you'd-ever-want-to-know documentation for software developers. (Even people who aren't software developers may get to see this stuff someday, but don't hold your breath. This stuff takes time. Production, you understand. Software developers are happy with mimeographed pages. Real people, Apple tends to believe, prefer nice bindings, pretty covers, and text laced with photos of hunks who normally do *Gentlemen's Quarterly* covers.)

Anyway, what computers were the writers doing their writing on? Apple IIIs, as a rule. But what word processing program did they use? Was it *Apple Writer III*? *Word Juggler*? Some other incredibly fancy, feature-packed word processing program?

Nah. Mostly they used the editor that's included with Apple Pascal. Just a clean, simple, modest little text editor. Nothing fancy.

AppleLine Lets Apples Participate in an IBM Network

Many large businesses rely on mainframe computers to perform enormous data processing tasks and serve as the hub of huge communication networks. In these businesses, IBM is by far the largest supplier of mainframe computers. Because many companies have an enormous investment in these existing systems, radical changes are impractical; companies are striving to evolve their information processing systems so that the impact on the organization is minimal.

What does this mean to vendors who would like to be suppliers of office automation equipment? Simply that their products should be able to coexist with and participate in an organization's information network.

One of the main components of an IBM network is a family of products known as the 3270. The IBM 3270 product line consists of a cluster controller and terminals. The cluster controller collects and interprets information coming from the mainframe computer and forwards the data to a terminal, where it is displayed to the end user. The controller also "concentrates" data being entered by multiple users in order to send them down a single communications link. The motivation for a cluster controller and terminal arrangement arose because communications lines were very expensive. Methods had to be developed to make maximum use of an existing communications line. The cluster concept was one such solution.

One of the most popular IBM cluster controllers is the 3274, which controls up to thirty-two terminals, also known as 3278 display devices. Anyone who has been near a corporation's main computer terminals will readily recognize 3278s by their green characters on a dark screen background. Large companies use these workhorses from the early days of office automation for everything from program development to order entry.

The proliferation of personal computers has dramatically affected the ways that terminals are used. For the price of what they used to pay for 3278s, users can now get multifunctional personal computers. After discovering the myriad uses of personal computers for local processing, users are now beginning to ask for better ways of obtaining information. For example, data entry is one of the most time-consuming tasks in spreadsheet analysis. If the information is readily available elsewhere, why not make it easier to put that information into a personal computer?

Vendors have responded to this need with a bewildering array of products that let users access information on remote systems using personal computers. These developments have, in turn, caused corporations to consider the purchase of personal computers instead of just terminals.

In large corporations where 3270s are quite popular, users are clamoring for cost-effective ways of connecting their personal computers to their IBM information network. One of the preferred alternatives is coaxial attachment to an IBM 3274 cluster controller, where personal computer interfaces are provided that allow one end of a coax cable to be attached to the IBM 3274 and the other to the personal computer. Since 3278s are connected this way, users can easily migrate from terminals to PCs.

Another advantage of using the coaxial attachment approach is that it is the least disruptive from an operations standpoint. In many cases, users simply unplug their 3278s and replace them with personal computers. With other alternatives, changes to the mainframe computer have to be made, sometimes resulting in unacceptable disturbances to data processing operations.

For the reasons just stated, Apple has taken the coaxial attachment approach in a product called AppleLine.

The AppleLine is a device that lets Apples attach via coax cable to an IBM 3274 cluster controller or its cousin, the 3276. With the appropriate software, computers using the AppleLine can emulate 3278 terminals, allowing users to obtain information from their IBM mainframes.

The AppleLine works with the Macintosh, Lisa, and Apple III personal computers. To take advantage of it, users will need communications software: *MacTerminal* for the Macintosh, *LisaTerminal* for the Lisa, and *Access 3270* for the Apple III.

Each AppleLine unit connects one Apple computer to a cluster controller. The AppleLine has two ports—an RS-232 port and a coax cable port. The AppleLine sits between the Apple and the IBM 3274 cluster controller and converts messages from asynchronous to 3270 coax protocols. One end of the coax cable plugs into the IBM controller, while the other end plugs into the AppleLine. The Apple computer in turn attaches to the AppleLine using RS-232 cable. The AppleLine is powered by a wall-mounted transformer that comes with the package.

When used with the AppleLine, Apple computers become truly multifunctional workstations, permitting full use of an Apple's local processing power, while at the same time allowing users to interact with IBM mainframes. One of the most dramatic examples of this is the combination of *MacTerminal* and the AppleLine.

MacTerminal utilizes advanced bit-mapped graphics and the power of Macintosh's thirty-two-bit processor to provide a powerful, easy-to-use communications program. The terminal parameters are set by pulling down a menu, selecting the options by moving the mouse, and clicking in the right boxes. There are no cryptic screens to decipher, since all selections are in simple English.

No spellers, no thesauruses, just the p-System editor.

We don't think they eat quiche, either, but we're still checking it out.

Boston Macintosh Society

Boston-based Macintosh owners can receive support, share information, and meet top hardware and software developers by joining the new Macintosh Users Group formed by the Boston Computer Society. The Boston Computer Society, which has about thirteen thousand members, is one of the largest user groups in the country. Although the club is located in Boston, one-third to one-half of all members are from outside the Boston area, says Macintosh Users Group president Jack Hodgson.

Members who join the Boston Computer Society can also join any of the society's forty special-interest groups and will receive a bimonthly BCS magazine and free subscriptions to any two special-interest group newsletters. Macintosh group members can dial into the main BCS bulletin board to exchange information with other members and to keep informed of the group's monthly meetings.

The June 13 meeting will feature a preview of MacPascal by Andrew Singer from Think Technologies. Afterward, group members will break into smaller groups to discuss Macintosh programming, word processing, and other subjects. Membership in the Boston Computer Society costs $24 a year. For more information, contact Jack Hodgson, Boston Computer Society Macintosh Users Group, One Center Plaza, Boston, MA 02108; (617) 354-7899.

To place *MacTerminal* in 3278 emulation mode, you simply select a terminal type labeled "3278." After setting the right communications characteristics, you turn on the AppleLine and enter the proper password; the Macintosh then becomes a terminal.

Regular users of 3278 terminals can take advantage of "function keys" that eliminate the need to type in commands. The user just pushes one key, which the mainframe interprets as if a complete command had been typed in. With other personal computers, emulating 3278 function keys becomes cumbersome because their keyboards are not designed to be only 3278s, so compromises have to be made.

Typically, multiple keystrokes generate the equivalent of a 3278 function key, making it rather inconvenient for the user to interact with mainframe applications programs. With *MacTerminal*'s superior graphics, a pull-down menu is used. This has two advantages: You don't have to enter multiple keystrokes and you don't have to take your eyes off the screen in order to search for the right combination on the keyboard. All the special function keys are made available on the screen. All you have to do is point.

Many professionals need to take work home occasionally. They can't bring a 3278 with them, but they can take a Macintosh and an Apple modem home and use the AppleLine's dial-in feature to continue their work. The user can access the network by emulating a 3278 terminal while at home, retrieve needed information, and use that information off-line with one of Macintosh's powerful business programs. This illustrates rather dramatically how Macintosh and the AppleLine make it convenient to access valuable information while you're far away from the mainframe and the IBM cluster controller.

The AppleLine and an Apple computer give users the benefits of personal computing and the advantages of 3270 communications at a reasonable price.—Arnie Lapinig

One from the Dealers

Bad news, good news—Apple dealers have heard both kinds lately. First, some bad news: Because of a shortage of Imagewriters, Apple is encouraging dealers to reserve the printers exclusively for Macintosh and to order other printers for the Lisa and the IIe and III.

Good news: Existing Imagewriter university contracts will be honored. Also, IIe and III owners *can* order fifteen-inch Imagewriters, since the Macintosh does not currently take advantage of the wider carriage.

Bad news: Because of unexpected demand, Apple will be totally out of stock on the Macintosh numeric keypad until August at least. Apple has advised dealers to stop taking orders for the product, explaining that such a backlog would be almost impossible to fill.

Good news: Apple will continue to bundle *MacWrite/MacPaint* software with each Mac sold through July 15. The promotional bundle was originally scheduled to be in effect only during the Macintosh's one-hundred-day introduction. After July 15, the *MacWrite/MacPaint* disk will cost $195.

The Basic behind the Basic

If you're going berserk waiting for Apple's MacBasic, you're not alone. If you need a MacBasic disk *right now*, we can't help. But if you merely want a glimmer of what this new language is all about, here's a suggestion. Get your hands on what's known informally as the "ANSI Basic specs."

ANSI is an acronym for the American National Standards Institute. You guessed it: They have a committee that got together and drew up a set of guidelines for a nifty, but totally hypothetical, Basic language. It doesn't exist, and may never exist, except on paper. A neat set of detailed suggestions, if you will.

Standards committees are something of a joke in the computer industry, as companies blithely go their own peculiar ways on everything from cables to keyboards. In this case, though, MacBasic author Donn Denman spent a lot of time poring over the ANSI Basic specs. Denman knows good ideas when he sees them. Denman, in fact, was a member of the committee that created the specs.

ANSI Basic is an "everything but the kitchen sink" language. The operative word here is *committee*. But many of the new (for Basic, anyway) features *are* implemented by MacBasic.

Infrared Secrets

Did we tell you yet about the cordless Mac mouse and cordless Mac keyboard yet? Last issue? The one before? Maybe not at all (we've got to start keeping a few back issues around here ...). Whatever, we keep hearing rumors about various cordless wonders being developed at Apple.

Which doesn't mean that we'll see either product released, of course. Maybe people don't want to type, or double-click, from across the room. IBM seems to think people go for this—people who buy dinky little computers, anyway. (The last comment heard was, "The Junior's infrared keyboard is great. It lets you get far away from the computer." Rude, rude, rude.) But Apple may not be as sold on cordless. Maybe it's just a way to keep the hardware people busy. They like to tinker, those hardware people.

What Mac Owners Want

A recent survey revealed that 20 percent of Macintosh owners want a letter-quality printer for their machines and 50 percent plan to purchase a carrying case, an Apple representative told members of the Macintosh User's Group in San Francisco. Macintosh owners who lug their machines—and printers—home from work every night probably aren't surprised at these findings; what is surprising is that Apple so underestimated the demand for carrying bags that dealers are discouragingly back-ordered. Mac owners in Los Angeles have found that the next best thing to haul their machines around in is the cardboard box that the carrying case came in—much less cumbersome than the Mac's original packings.

The same Apple representative also estimated that 512K Macs will be available by the end of the year and that double-sided drives will not appear until 1985. Laser printers are in prototype form at Apple, and the company estimates that five hundred software packages will be available by the end of 1984.—Marc Benioff

Big Bauds

The two serial ports on Macintosh are fast. Just how fast is explained with a lot of numbers—one megabit per second (if clocked externally) is a phrase often heard.

One of our researchers came up with a different way to state the speed question (actually, he found this in the vast *Inside Macintosh* documentation). This may, *may*, be more meaningful to readers acquainted with modems and baud rates. Everyone else may get as much out of phrases that include "clocked externally."

Typical modems operate at maximum speeds of either 300 baud (fairly slow) or 1200 baud (a delicious increase in speed—zippy!). The Macintosh serial ports operate at both those speeds and a few more, besides.

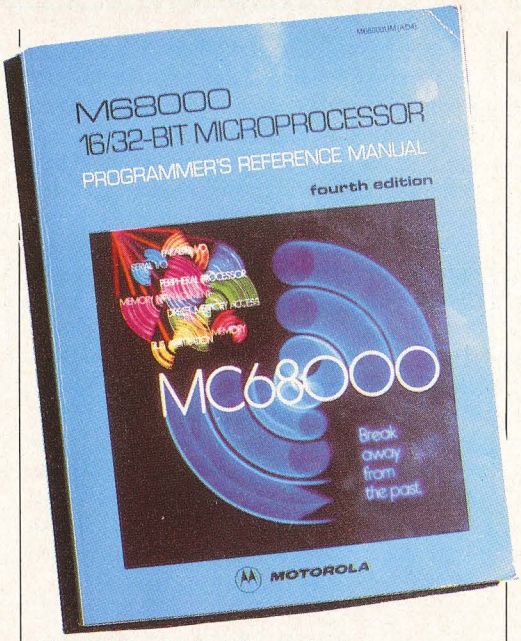
Depending on what assembly language instructions you pump in, you get back baud rates like *300*, *600*, *1200*, *1800*, *2400*, *4800*, *7200*, *9600*, *19200*, and *57600*.

57600 baud! If you've ever experienced 1200 baud, 57600 must sound faster than light (but probably isn't).

Only one of the two serial ports (Apple calls them Port A and Port B) makes high speed easy, however. Here's the relevant paragraph from Apple's internal documentation:

"Port A now has an added feature: It may be used simultaneously with disk accesses at the highest async baud rate with no worry of overrun. The disk driver now polls Port A whenever it must turn interrupts off for longer than 100 microseconds, and then passes any acquired data to the async driver. SCC channel B should be used for output-only connections such as to printers, or at low baud rates (a 300-baud modem, for instance)."

We're glad that's cleared up.



Free Motorola Manual

A good book on the Macintosh and Lisa processor is the *M68000 16/32-Bit Microprocessor Programmer's Reference Manual, Fourth Edition*, a detailed guide to the Motorola 68000. Its contents include architectural description, data organization and addressing capabilities, instruction set summary, exception processing, and other information pertinent to the operation of the computer. Any assembly language programmer will need this information to program the computer effectively. This book is available from Motorola at no charge. Contact Motorola at 4030 Moorpark Avenue, Suite 121, San Jose, CA 95117; (408) 985-0510.—Marc Benioff

Coming Soon(er or Later)

There's probably a hierarchy lurking around within unreleased Macintosh software. It goes something like this: Software we've seen and know is coming, software we haven't seen but know is coming, software products that are announced but seem shaky (speculation involves rough judgment calls), and rumors.

The second category is the largest. There's a long list going around the office of everybody who's developing software for Macintosh.

But consider these rumors: There's a marvelous expert system rumored to be under development for Macintosh. Something to do with double-clicking on maps. A "visible assembler" where you can watch the contents of the 68000 change as your program is running (quite educational). A few incredible free-form database management programs (no need to worry about files, records, and such) and a word processor with a database built in. At least three programs to make Mac speak up, whether while reading *MacWrite* files, programs, or merely telling you the news from Dow Jones in the morning (good speech quality, it's rumored). We're waiting for a call about a biofeedback program, complete with headset, or wristset, or something,

Truthfully, there's only one thing we'd like better than having all that software on sale today: ads for the software. Ads. Money. Writers. More pages. Great paper. Forty-hour weeks. Better Miscellanea....

Hot Tickets to the Mac Show

Apple's offer of half-price Macintoshes to students and faculty of the University of Texas has resulted in such demand for machines that a form of option trading has developed for "computer tickets" that allow purchasers to order machines, according to an article in the *Austin American Statesman*. About six thousand such tickets were distributed in March by the Micro Center, a computer store selling the Macintosh under the UT-Apple University Consortium agreement. The tickets are being given out on a first-come, first-served basis to students and faculty at no charge and give the date and time that a ticket holder can order a machine. Ads have been running in the student newspaper, *The Daily Texan*, offering to sell tickets for as much as $65.

Although, under its agreement with the university, Apple will offer an unlimited number of machines for sale for three years, students must be enrolled full time both when the computer is ordered and when it is delivered. Graduating seniors and students who will be out of town for the summer have created the greatest demand for the tickets.

About one thousand students, staff, and faculty members have purchased Macintoshes so far under the UT-Apple agreement, according to the *Statesman*. Demand quickly outstripped supply in March when Macintoshes were first offered for sale. The university attributed the shortage to the fact that "the Apple factory is not up to full capacity yet and will not be until late summer or early fall. This means that Apples will be in short supply until then."

Digitized Mac Art

This summer, Macintosh owners will be able to hook up a small camera to their computers and see digitized pictures on the screen. The MicronEye is a device that connects to the Mac and acts as a simple digitizer. Point it in one direction and a digitized image of that area appears on the screen. Scan it and see images that you point to scroll by. Users of the MicronEye will also be able to save their pictures to disk. The product is currently available for the Apple II and Commodore 64 computers and retails for $295. A price and release date for the Mac version of the MicronEye have not been set. The device will be available from Micron Technology, 2805 East Columbia Road, Boise, ID 83706; (208) 383-4000.—Marc Benioff

Mac-Slots Beats Vegas Hands Down!

The action is fast. The excitement is real. And until now, there's never been a game for the Macintosh like Mac-Slots.

TWO Complete Games in One Package.

A coin operated slot machine. Even better than the original "one armed bandit."

Keno. Non-stop action like you've never seen it before.

Both games are totally mouse-controlled and let you pit your skill and luck against real Las Vegas odds.

It Turns Your Macintosh Into a Casino.

Mac-Slots has all the glitter and competition of the real thing, with some surprises!

Hit the jackpot and the Pit Boss puts in an appearance.

Extend your credit with the Cashier — or cash in your winnings.

Order Cocktails from the bar. But watch out! They'll affect your play.

Need a place to hide? Break for the Mac-Slots Restroom and find out why "it's different."

So if you want an experience that'll beat Vegas, Reno and Monte Carlo at their best... Put your money on Mac-Slots.

Both Games for Only $77.77. AVAILABLE NOW!

CAUTION: USE OF THIS SOFTWARE FOR ACTUAL WAGERS IS ILLEGAL IN MOST STATES.

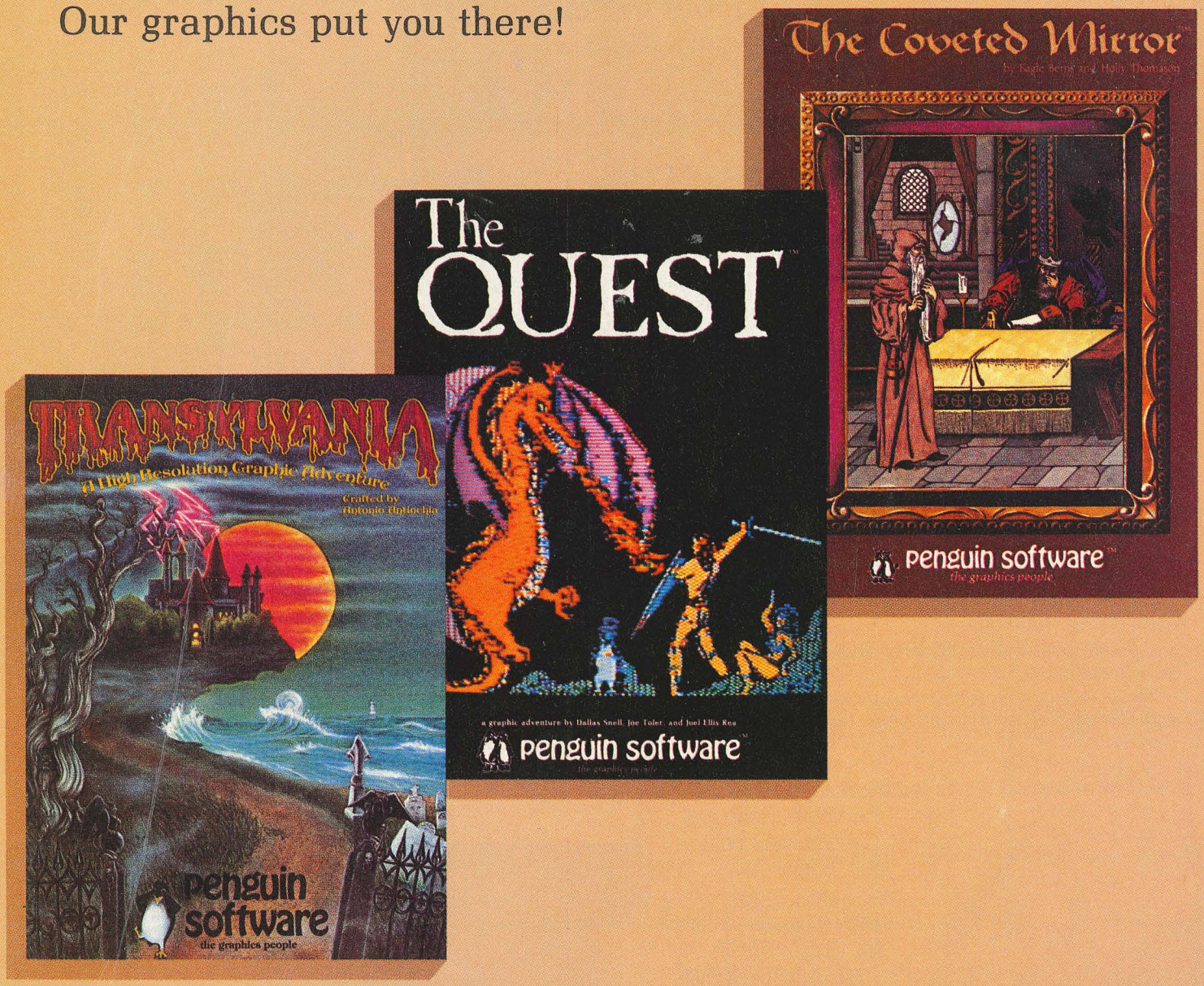
To order, call **Toll-Free: 1-800-235-6646, Extension 561** • In California: **1-800-235-6647, Ext. 561** • In Los Angeles: **213-660-7940, Ext. 561.** MasterCard and VISA accepted. Or see your local software dealer.

Mac-Slots

Send Check or Money Orders to: **Soft-Life Corporation, 2950 Los Feliz Boulevard, Los Angeles, California 90039 (213) 660-7940.** Dealer inquiries invited. Add $2.00 shipping and handling. California residents add State sales tax. Marketed solely by Soft-Life Corporation. A product of DataPak Software.

ADVENTURE

Our graphics put you there!



Thoughtful prose and stunningly detailed graphics by some of the best computer artists bring life to the Penguin Software adventure series. Whether in the beautiful detail of Transylvania, the expanse of The Quest, or the animation throughout The Coveted Mirror, our graphics make you feel like you're there, experiencing the adventure as you play.

The Penguin adventures are or will be available on disk for the Apple II series, Atari home computers, Commodore 64, IBM PC, and Apple Macintosh.
Double-sided 5¼″ disks are $34.95, and Macintosh disks are $39.95.
Graphics created with The Graphics Magician.

penguin software™
the graphics people

830 Fourth Avenue
Box 311
Geneva, IL 60134
(312) 232-1984