The Developer Business Report *October 1993*

Apple Directions

Inside This Issue

PowerPC Details in Next Month's Mailing

The November Developer Mailing will include a wide range of information on the upcoming PowerPC processor–based Macintosh platform, including an in-depth white paper, "PowerPC Technology: The Power Behind the Next Generation of Macintosh Systems." Keep a close eye on your mailbox in mid-October and don't miss this important information!

Apple News

System 7 Product Line Expanded

In early October, Apple will expand the System 7 product line to better meet the needs of an increasingly diverse Macintosh customer base. Customers will be able to select from two "flavors" of System 7—a base version and a full-featured version. The base version will continue to be called System 7, as it is today; the new full-featured version will be called System 7 Pro.

Driving the expansion of the System 7 product line is a simple reality: One size doesn't fit all anymore. For example, a 12-year-old student using a Macintosh Classic® to write English essays at home has very different needs than a business professional whose Macintosh is the focal point for all of his/her work. The student needs a baseline version of System 7 that will allow him to do word processing and run a few games. The business person, on the other hand, needs a more powerful system allowing her to communicate via fax and electronic mail, track contacts, automate expense report processing, collaborate across project teams, and, perhaps, work with multimedia documents.

With the introduction of System 7 Pro, widely disparate customers like these will be able to more closely match their system software choice with their specific needs. System 7, the base product, will be the same System 7

Please turn to page 4

Strategy Mosaic

Making Newton Software that Sells

By Gregg Williams, Apple Directions *Staff*

Everything you know is wrong.

About the Newton MessagePad, that is. Newton devices are no more computers than pads of paper or microwave ovens are. If you think of a Newton device as a computer, that's understandable—but you'll definitely make the wrong decisions for you and your company. Here are some key ideas that should shape the overall direction of your first Newton software products. (Note: Throughout this article, I use the term *Newton* generically; the points I make apply *today* to the just-released MessagePad communications assistant and *tomorrow* to forthcoming devices, from Apple and others, based on Newton technology.)

WYSIWYG is short for "what you see is what you get"—and it has been the dominant paradigm in this industry. This approach has worked; it gave many people (who wouldn't have touched a computer otherwise) the confidence to use a computer.

WYSIWYG, which demands that the image on the screen match its printed result, also dictates that the method of entering data closely correspond to the data's final presentation. It's true that this approach works—but is it necessary?

Please turn to page 3

AppleDirections

Volume 1, Number 5

Apple Directions, Apple's monthly developer newsletter, communicates Apple's strategic, business, and technical directions to decision makers at development companies to help maximize their development dollar. It is published by the Developer Support Information group within Apple's Developer Press.

Editor
Paul Dreyfus (AppleLink: DREYFUS.P)

Technical Editor
Gregg Williams (GREGGW)

Business & Marketing Editor
Dee Kiamy (KIAMY)

Designer
Robert Stone

Production Editor
Lisa Ferdinandsen (LISAFERD)

Contributors
Juan Bettaglio, Mark Bloomquist, Alex Dosher, Tim Enwell, Stacy Moore, Rilla Reynolds, Caroline Rose, Adam Samuels, Tracy Swedlow, Anne Szabla, Wendy Tajima, Jessa Vartanian

Manager, Developer Press
Dennis Matthews

Manager, Developer Support Information
Greg Joswiak

Production Manager
Diane Wilcox

PrePress/Film
Aptos Post

Printer
Wolfer Press Co., Inc., Los Angeles, CA

Editor's Note

Computer, Read My Lips

Speech recognition is here—*usable, useful* speech recognition in the form of Apple's new PlainTalk speech recognition/synthesis technology. Even better, this technology is available to any program that wants to use it . . . and I'm leaving fingernail tracks in the dirt, trying to keep from buying a PlainTalk-capable Macintosh. If you haven't heard Macintosh speech recognition, I urge you to find one of Apple's new AV technology computers (the Macintosh Centris 660AV and Macintosh Quadra 840AV) and try it out.

The only Macintosh Centris 660AV I've seen so far is my upstairs neighbor's proud new possession. (He's *not* an Apple employee—in fact, he does Windows for a living.) The first thing I said to it was, "Computer-whattimeisit?" (not "Computer—what—time—is—it?") and it told me, using the also impressive speech synthesis part of PlainTalk. My neighbor asked the same thing, and *he* got the same answer.

Two amazing things about this: First, the computer recognized what two different people said, without either of us training it to our voices. This is called *untrained, speaker-independent speech recognition,* something that's never been done commercially (at least, not on a computer costing well under $2500). Second, this is also *continuous-speech recognition,* another Holy Grail of speech recognition; PlainTalk does not have to have silence around each word to recognize it. Put them both together, and you have a key cliché of every science-fiction movie you've ever seen—and it's just jumped off the screen and into your lap!

Some caveats: Obviously, this first implementation of PlainTalk isn't an unbounded solution—it can't take dictation, it *doesn't* understand what you say, and it recognizes unaccented English only. But, out of the box, it can recognize and automatically execute any menu command or dialog-box button the user speaks. (This works for existing, unmodified Macintosh applications.) It also responds to over 20 types of commands and utterances, including "What day is it?," "Close window," "Switch to [name of another application]," "Open [control panel]," "Shut down," and "Print from page [x] to [y]."

In addition, there is a new folder in the System Folder called Speakable Items. (I'm lobbying for adding another folder called "*Un*speakable Items"—maybe next April. . . .) If the user speaks the name of any icon in this folder, system software will attempt to open or execute it. Since the icon can be an application, an alias, a CE Software QuicKeys file, or an AppleScript script, the number of things that you can trigger with a voice command is almost unlimited. (And don't forget, the more Apple events your application responds to, the more controllable it is by AppleScript and, thus, by voice.)

PlainTalk can do all the above things on its own (or with a little help from its user). What can you, the developer, do in addition? You can create applications/aliases/scripts for the Speakable Items folder and ship them with your product. If you really want to make your product state-of-the-art, you can add what are called speech rules to your application. (Speech rules compile into a resource that you add to your application or have the user drop into the System Folder.) Speech rules let you add more powerful voice control to your application than you (or the user) could with macros and scripts.

So what are speech recognition and synthesis good for? Lots of stuff, including "busy hands, busy eyes" situations (where the user is away from the keyboard, mouse, or monitor) and applications that normally make heavy use of the mouse and keyboard. Users with hand and arm problems may also benefit from applications that substitute speech recognition for keyboard and mouse activity. Together, speech recognition and synthesis will fundamentally change how people work with their computers.

PlainTalk speech recognition is currently available only on the AV Macintosh computers, but it will also be available on Apple's PowerPC processor–based line of computers. Now is the time to add speech recognition—where it makes sense—to your products. Speech recognition will become an integral part of the Macintosh experience, and soon.

Gregg Williams
Technical Editor

For Partners and Associates Only

October 1993

Apple Directions

Apple Enhances Developer Support

To support Apple's broad range of new technologies and to respond to developer feedback, Apple's Developer Support Center has just announced a series of enhancements to its U.S. developer support activities, some oriented to new technology and product development opportunities, others enhancing existing support programs.

To sum up the changes, Apple is improving service across the board:

- The Associates Program remains an affordable, self-support program, while some of its individual services have been improved.
- We've added a new program for strategic PIE (Personal Interactive Electronics) developers.
- We're expanding the Apple Partners Program to support PowerPC technology and new Macintosh system extensions, and we're changing the price and giving it a new name: the Macintosh Technology Partners Program.
- We've integrated the Apple Multimedia Program with the other developer programs.

According to Ike Nassi, vice president of the AppleSoft Development Products Group, "Apple pioneered great developer support. The goal of these enhancements is simply to make the industry's best service better by supporting new products and technologies, and improving the mix of services we offer to our development partners."

Here's a look at the prominent changes.

Basic Services: New and Improved

The Apple Developer Mailing has undergone significant improvements over the past six months. This past April, the popular Developer CD Series was redesigned to provide 1.8 gigabytes of development tools, documentation, and system software each quarter. In the past, we had been constrained by space and organization to deliver only a broad sampling in these areas with relatively little new information each month. This information is now delivered on three separate CDs, one each month, with each CD covering a specific area in depth. The format has also been improved to make the information easier to retrieve with Apple's full-featured document browser, DocViewer.

In June, we refocused the content of *Apple Directions* to provide more timely coverage of insights into Apple's business and technology strategies. We also redesigned the format to be easier to read, handle, and store.

Both the Developer CD Series and *Apple Directions* now include information on all Apple technologies, including PIE, multimedia, PowerPC, and Macintosh technologies. The Associates Program, which continues to offer the Apple Developer Mailing and more, will remain at the annual subscription price of $350.

New PIE Partners Program Introduced

Apple's Partners Program, since its founding in early 1989, has been dedicated to the needs of strategic developers for the Macintosh. With the delivery of the Newton MessagePad and Toolkit, another group of developers is in need of Partner-level support. To meet the demand, we've instituted a new program, the PIE Partners Program.

The Developer Support Center will coordinate the new PIE Partners Program with Apple's other developer programs; the PIE Partners Program itself will be managed from within Apple's PIE organization, which will provide tools, documentation, and other services for developers. PIE Partners will receive programming-level development support directly from PIE engineers on Newton and other emerging PIE technologies. The annual fee for qualified participants to become a PIE Partner is $2850.

Partners Program Becomes Macintosh Technology Partners Program

Strategic developers focusing on Macintosh technology will now receive expanded support covering all new Macintosh technologies—including PowerPC technology and new system extensions such as QuickDraw

Please turn to page B2

AppleDirections

Volume 1, Number 5; Supplement

Apple Directions, Apple's monthly developer news- letter, communicates Apple's strategic, business, and technical directions to decision makers at development companies to help maximize their development dollar. It is published by the Developer Support Information group within Apple's Developer Press.

Developer Support

continued from page B1

GX and PowerTalk—through the Macintosh Technology Partners Program. Our development support focus will match Apple's top-priority technologies so developers can hit the market quickly with new innovations. The Macintosh Technology Partners Program annual fee is now $1500. All Apple Partners will continue to get all the Associates features included in the annual fee.

All existing Partners will pay no additional fees until their annual renewal date, at which time they will become members of the Macintosh Technology Partners Program. Developers qualify for the Partners Programs by demonstrating their involvement in work that supports Apple's long-term product and business strategies.

"As Apple's technologies evolve, the developer programs need to keep pace," said Shirley Stas, Apple's Developer Support manager. "In order to continue providing top-quality support on our increasing number of technologies, we are finding it necessary to increase the fees for one-on-one technical support."

Apple Multimedia Program

In response to developer feedback, the Worldwide Multimedia Group has joined forces with the Developer Support Center to integrate their programs. Developers will find it easy to join the Apple Multimedia Program and receive essential information focused on the concerns and needs of the multimedia community. The successful Apple Multimedia Program was launched last October and has already attracted 500 members. This program is an invaluable source for Apple multimedia developers and enthusiasts. The Apple Multimedia Program is offered to all Associates and Partners for an annual fee of $300.

Apple has long understood that its success depends on providing, at a reasonable cost, the platforms, tools, and assistance developers need for doing their jobs. Apple's developer programs will continue to do just that.

"A quick look around the industry will show that Apple provides the best support at the best prices. In fact, with the new features and additions, we think the best support program in the world just got better," said Nassi. "Please continue to AppleLink your comments and recommendations to DEVFEEDBACK. With your help, we will all stay on top."

More information about the programs is included in the October Apple Developer Mailing. If you are interested in signing up for one of these new programs today, contact the Developer Support Center at (408) 974-4897. ♣

Strategy Mosaic

Newton Software

continued from page 1

Consider the classic expense account form. Each page represents a week and gives you a spreadsheet of numbers with each day having its own column and each expense category having its own row. Even sitting at your desktop computer, you're lucky if you have a monitor big enough to show the whole form at once.

Now consider the following alternative. You have a PDA (personal digital assistant) about the size of a paperback book—it has to be small enough for you to take it with you everywhere. When you pay for your breakfast, you open your PDA, write in the amount , and close it. From its internal clock, the PDA knows what day and time it is. From that, it infers that the amount you entered must have been for breakfast and categorizes the expense accordingly. After your trip, you connect your PDA to a printer, and it prints the expense account document.

How long did it take for you to add your breakfast to your expense account—five seconds, maybe ten? With this example, you begin to understand that a PDA *isn't* a computer—and that, for the right tasks, a PDA is *better* than a computer.

[Author's note: In last month's column, "Why Newton Will Succeed and How You Can Get Started," I wrote about the Newton customer and how Newton developers will interact with Apple. Among the people I talked to for last months' and this months' columns were Philip Ivanier, manager of development relations, and Ken Wirt, director of marketing for Apple's Personal Interactive Electronics Division.]

Here is a very important paradigm shift. The approach just described is definitely *not* WYSIWYG—the way you enter data is different from the way you finally want to display it—*and it works.* This leads me to an important observation: Don't try to port your Macintosh application to the Newton—it *won't* work.

Apple sees the biggest opportunities for Newton titles as being in the following areas:

- business
- entertainment
- reference
- self-improvement
- sports

I'm loosely using "titles" to mean any software that users will put in their Newton devices, both data-heavy—for instance, maps—and computation-heavy—such as traditional Macintosh applications. In general, Newton titles will be more "data" than "program," but the main advantage to a Newton title will be the *intelligence* that the Newton provides in allowing rich, creative, easy access to (and manipulation of) that data.

Evaluating Product Ideas

Having touched upon the *what* (what categories of titles to think about), let's next look at the *how.* In this case, *how* translates to "How will Newton customers use their Newtons devices? What will and won't they be using their Newton devices for?" If you ignore these questions, you will market the world's best spreadsheet for the Newton—and it will fail.

Last month, I said that Newton customers share several important traits:

- They want to have a PDA with them all the time.
- They are always moving from place to place.
- They need to keep track of many small, miscellaneous pieces of data.

Here is another set of useful observations. When anthropologists observed potential Newton customers at work and at play, they generalized that these people do the following three things:

- gather data
- organize the data they have
- communicate with others

You should keep these behaviors and usage profiles in mind. When you're brainstorming new products, see if each idea is consistent with the findings about customers. If it isn't, the idea may not be a good one.

It may also help you to keep the following phrases in mind: *in touch* and *in the moment.* The phrase *in touch,* according to Philip Ivanier, applies on more than the personal level of being in contact with other people. According to him, Newton allows individuals "to better stay in touch not only with other individuals, but also with their company and information in their desktop systems."

Data transfer between your Newton and your personal computer should be so easy that you see yourself as being "in touch" with your desktop computer. *In touch* also means letting users take care of items immediately so that they don't need to even start a to-do list.

For example, a user may jot a note down, then tell Newton to fax it to somebody. Newton takes care of the details, and the user feels "in touch" with the fax recipient and can go on to other work.

Industry Watch will return next month.

Of course, these examples are also "in the moment"—these people are getting small tasks done, spontaneously, and are then able to forget about them.

Making a Good Newton Product

I've said it before, but it bears repeating: Don't, don't, *don't* try to shoehorn your current product line into the Newton. Even if a product makes sense when ported directly (a simple game, for example, or an address book), you can probably find ways to improve it.

Computer programs usually do many things; often, too many things, judging by the size of instruction manuals these days. However, Newton titles will often do one thing well, and they will go out of their way to make the user's work easier.

For example, recall the expense-account example at the beginning of this column. This hypothetical program didn't make the user enter the date or the name of the meal. All the user did was enter $11.38 in a box labeled *Meals,* and the Newton entered the data internally as "breakfast, San Francisco, August 11, 1993, non-entertainment meal, $11.38." Simple, and it makes the user's life easier.

Apple Directions On Line for November

The November issue of *Apple Directions* will be available on AppleLink as follows:

September 1—Preliminary draft copy
September 15—Final copy

To view *Apple Directions* on line, follow the AppleLink path Developer Support:Developer Services:Periodicals:Apple Directions:Apple Directions November.

Another characteristic of good Newton software is that it lets users personalize their data. This can mean many things—sorting a list on a certain field, scribbling on a piece of prepared data (a map, for example), changing the default typeface. We've seen it on the Macintosh; on the Newton, look for both fun and useful ways of letting users personalize their data.

Though I believe you can teach an old dog new tricks, the dog will initially do better with the ones it already knows. In other words, when searching for a new Newton product to develop, start with what you already do best. Let's face it, the Newton market will be full of variables—at least while you're starting out, why risk adding one more? Start with what you know.

Finally, to design a good Newton product, keep the worldwide market in mind. Newton devices are very language-specific, but Apple will soon be coming out with Newton devices localized to other languages, including German, Japanese, and French (in the near-term). According to Ken Wirt, a commercial product has a better chance of succeeding if it makes sense in two of the three Apple "geographies"—U.S., Pacific (including Japan, Canada, and Australia), and Europe. For example, you may be able to develop a business title that people in the United States and Europe might use. This would be better than a title that works only in the Pacific countries, which conduct business differently than the United States and Europe.

Of course, there are endless variations on doing business, and Newton suggests some of its own business opportunities. Here are some that have come to mind here at Apple:

- Even if you primarily provide services to your customers, you may want to sell them Newton hardware and software as well. If you can sell them a program they find useful that helps them do business with you or that feeds information back to you, the indirect profit may be even more important than the direct profit from the sale.
- Many companies of various sizes may want custom Newton software with which to run their businesses. If you do Newton programming, you may want to pair up with companies that need your services.
- Repackage someone else's data. Think of the five categories of Newton titles: business, entertainment, reference, self-improvement, and sports. Find some interesting data, then think of a new way to package and add value to it.
- Because Newton software is intrinsically fluid and modular, there will be a market for add-on products that improve upon other Newton products. If your improvement is good enough, you'll have a market for it.
- For many (but not all) users, the Newton will not live in a vacuum; instead, it will live in a web of computing power and networks. People will always be interested in Newton products that help them connect to their computers—and to other sources of information—in meaningful ways.
- Again, one exciting possibility for the Newton is using it to gather data in the field, getting it back to the home office (or wherever), and putting the data to some use. I have a feeling that someone will use this idea to make major cost savings or generate an entirely new, untapped source of revenue.

Beginning Now

The Newton MessagePad kicks off the industry of personal digital assistants. It's a totally new industry with new rules, most of them yet unknown. Though companies with more money usually have an advantage, no one is an expert yet, and a small company with a few good ideas can make its mark. If you're tired of fighting the industry standard and the installed base, here's where you have your best chance of success.

As the months go by, you'll see more Newton titles come out and more Newton devices—not just from Apple but from Sharp, Matsushita, Motorola, and Siemens/ROLM. Make dazzling software. Offer me things I won't want to live without. Change my life—or at least make it easier or more fun. I'm in a buying mood, and I'll bet I'm not the only one. ♣

Apple News

System 7 Pro

continued from page 1

available on store shelves today, providing the basic tools to work smarter and faster.

System 7 Pro, the new full-featured product, will include everything in the base offering plus three bundled extensions: PowerTalk (AOCE collaboration services for the individual), AppleScript, and the latest version of QuickTime. System 7 Pro offers customers the tools to streamline their work and communicate more effectively as well as offering a rich foundation for creating integrated, customized solutions.

We anticipate that many of today's System 7 users will upgrade to System 7 Pro to take immediate advantage of the functionality made possible by our latest technologies, PowerTalk and AppleScript. We also expect that over time, base System 7 users will naturally upgrade to System 7 Pro as their needs expand. For example, if the 12-year-old student wanted to send his English essays to his teacher by fax, or use an on-line service, he would likely upgrade to System 7 Pro.

For customers, System 7 Pro will be the place to get the latest and greatest Apple technologies in one complete package. What this means for you is that there will be an immediate and expanded market for products that take advantage of new technologies, because more customers than ever before will be capable of putting your products' added functionality to work. By making these powerful extensions part of our mainstream system software offerings, Apple intends to rapidly drive their penetration in the installed base, expanding the market for your full-featured products and helping you differentiate them against the competition.

We strongly encourage you to check your current applications for compatibility with the soon-to-be-released System 7 Pro. The most current release, Beta 11, is available on this month's Developer CD for testing and development purposes. Aside from compatibility testing, we encourage you to speed up your plans for adopting PowerTalk, AppleScript, and QuickTime technologies, if you haven't done so already. Customers selecting System 7 Pro will be demanding applications that take advantage of these leading edge technologies, and you don't want to miss out on this significant market opportunity!

Look to the *Apple Directions* November issue for in-depth coverage of Apple's expanded system software product line, including target markets, positioning, pricing (including upgrades), and ideas on how to best capitalize on System 7 Pro as you adjust your own business and product strategies.

Apple News

Multimedia Tools

Apple intends to be a leader in cross-platform multimedia technology, and its goal to sell between 1 and 1.5 million CD-ROM drives means that there will be more money to be made in CD-ROM products.

Apple recently announced three new multimedia upgrade kits, the Apple CD Multimedia Kit (for Macintosh computers), the Performa CD Multimedia Kit, and the CD Multimedia Kit for PCs (for MS-DOS/Windows computers). These kits, which Apple is aggressively marketing through various channels, include an AppleCD 300 CD-ROM drive, a pair of AppleDesign Powered Speakers, and multiple CD-ROM titles.

The Performa CD Multimedia Kit offers Performa owners a simple, all-in-one package. The AppleCD Multimedia Kit and the CD Multimedia Kit for PCs let the customer order by mail up to 3 free CD titles from a list of 30 business, reference, education, and entertainment titles.

If the CD-title market looks attractive to you, you need to start working on those titles now. Here's a list of the multimedia tools and services that Apple currently offers. (For APDA ordering information, see page 23. Prices listed are for customers in the United States only; if you're outside the United States, call your local Apple office for prices in your area.)

Apple Media Tool and Programming Environment

Introduced at Boston Macworld in August, the Apple Media Tool is the first of two products that provide a comprehensive multimedia authoring solution for both the Apple Macintosh and Microsoft Windows platforms. The Apple Media Tool is a no-programming, direct-manipulation tool that allows nontechnical users to create interactive multimedia titles that include text, sound, PICT graphics, and QuickTime movies.

The Apple Media Tool Programming Environment, just released by APDA, is an object-oriented language and application framework that allows programmers to add specialized features to a project created with the Apple Media Tool.

The output of the Apple Media Tool is a player file that can be executed by either a Macintosh or a Windows runtime program. The Apple Media Tool thus allows you to create a single multimedia title for both platforms. In fact, you can even create a single CD that will run correctly on both Macintosh and Windows computers.

The Apple Media Tool can also output a project in a text format. This source code, meant to be manipulated by the Apple Media Tool Programming Environment, is forward-compatible with the upcoming ScriptX multimedia authoring language from Kaleida Labs. This compatibility will make it easy for you to move existing Apple Media Tool projects to other hardware platforms that will support ScriptX.

The Apple Media Tool is now available from APDA (APDA order #B1440LL/A) at a special price of $995 until September 30, after which it costs $1495. The Apple Media Tool Programming Environment (order # R0535LL/A) sells for $2995. A product containing both the tool and the programming environment, The Apple Media Kit (order # B1453LL/A), is avilable for $3495. Like all APDA products, they come with a one-year, unconditional refund policy.

QuickTime Developer's Kit

QuickTime is Apple's architecture for time-based media—video, animation, and sound. QuickTime 1.6 offers significant improvements over version 1.5, including faster playback of existing movies of twice-normal size; a new Compact Video compressor, which can run movies two to four times larger than before; a much smaller memory "footprint" than the previous version; support for Sound Manager 3.0 and Kodak's Photo CD; and better movie playback, especially on 1-bit PowerBook computers.

The QuickTime Developer's Kit includes QuickTime 1.6, documentation, code headers, and sample code. The QuickTime Developer's Kit version 1.6 (order # R0147LL/C) is $195. Upgrade kits are also available for QuickTime 1.0 (order # R0456LL/B, $99) and QuickTime 1.5 (order # R0533LL/A, $25).

QuickTime for Windows Development Kit

This product contains all the software, code, and documentation you need to add QuickTime support to Microsoft Windows applications (order # R0453LL/B, $195). QuickTime for Windows 1.1 is implemented as a set of Windows Dynamic Linked Libraries (DLLs) and supports Windows Media Control Interface (MCI) and Object Linking and Embedding (OLE).

Apple Multimedia Program and Information Mailing

Apple has taken the Apple Multimedia Program and turned it into an optional program available to anyone participating in the Apple Associates and Partners Programs in the United States and Canada. This program will provide Apple Associates and Partners with resources, third-party discounts, and special information on AppleLink; it costs $300 per year. For more information, contact the Apple Developer Hotline at (408) 974-4897 in the United States.

If you're located outside the United States and Canada, or if you're not participating in the Associates and Partners Programs, you can obtain the materials (including software, videotapes, and printed publications) sent to members of the Apple Multimedia Program by ordering the Apple Multimedia Information Mailing from APDA (order # R0494LL/A for PAL-format videotapes; order # R0495LL/A for NTSC).

Author's Solution for Interactive Electronic Books

This tool helps you create multimedia "books" that include text, graphics, sound, video, animation, and interactivity. Author's Solution for Interactive Electronic Books (order #R0493Z/A, $350) includes the Voyager Expanded Book Toolkit and a booklet containing material from Apple's book *Demystifying Multimedia.* Since the Voyager product is based on Apple's HyperCard®, you can add and modify HyperTalk® scripts to extend what your multimedia book can do. You don't have to do any programming to create an electronic book with this tool, though. (In the October 1993 APDA catalog, the entry-level version of this product, Author's Solution for Interactive Electronic Books, Level I, is not available.)

PowerPC Development Tools Announced

At the Macworld Expo, held in August, nine leading Macintosh development tools vendors announced plans to provide tools and languages for the next generation of Macintosh computers based on the PowerPC microprocessor. These tools will augment the

Macintosh on RISC (formerly the Macintosh on PowerPC) Software Development Kit that Apple Computer, Inc., plans to begin limited seeding to developers in the fall. (See "Making the Transition to PowerPC" on page 14 of the August 1993 *Apple Directions*.)

The tools provided by Apple and other companies are being released to help you create native applications that tap the power and performance of the new PowerPC processor–based Macintosh computers, expected to begin shipping in the first half of 1994.

Announcing their support were the following companies:

- Absoft Corporation, the developer of MacFortran II, whose PowerPC tools will include Absoft FORTRAN 77, a C/C++ compiler, the Fx multilanguage debugger, and the Cray/Absoft Fortran 90 compiler
- Bowers Development Corporation, the developer of AppMaker, an interface builder and code generator for the Macintosh
- Echo Logic, developers of FlashPort for Macintosh, a binary software translation tool that enables you to convert 680x0-based Macintosh applications so that they run in native mode on the PowerPC processor–based Macintosh
- Language Systems Corporation, developer of LS FORTRAN and LS Pascal
- Metrowerks, Inc., the developer of several tools for PowerPC processor–based Macintosh computers, including integrated development environments for C, C++, and Pascal, and PowerPlant, an object-oriented application framework
- MicroAPL, the developer of PortAsm, a tool that translates 680x0 assembly language to PowerPC assembly language
- Sierra Software Innovations, the developer of Inside Out II, a multiuser relational Macintosh database engine; p2c, a translation tool for converting Pascal and Object Pascal to C and C++; and IcePick, the MacApp view editor
- Symantec Corporation, the developer of THINK C and THINK Pascal; the company is also working closely with Apple on a next-generation, native development environment for PowerPC processor–based Macintosh systems
- TGS Systems, developer of Prograph, a visual, object-oriented, dynamic programming language and environment

Collectively, PowerPC tools provided by Apple and these vendors will support a range of programming languages and development tools, including compilers, debuggers, translation tools, and interface building tools. "Our developers are anxiously anticipating PowerPC and want to take advantage of it as soon as possible," said Ike Nassi, vice president of Apple Development Products in the AppleSoft Division. "Our top priority is to present the smoothest migration path possible to PowerPC for all our developers by ensuring the availability of a wide range of tools to meet their varying needs."

Apple is concentrating on providing tools for writing applications in C and C++. These Apple-provided tools will include the Macintosh on RISC Software Development Kit, a cross-development environment enabling the development of native PowerPC applications using a 680x0-based Macintosh.

Apple is supporting third-party tools vendors in the creation of complementary tools for developers using languages other than C and C++, such as assembly language, Pascal, Object Pascal, and others.

Over time, Apple plans to offer a full range of products based on PowerPC technology, from entry-level to high-end Macintosh personal computers. The computers will ship with Apple's standard operating system, System 7, and are designed to offer developers and customers high performance and a variety of new features at competitive prices. In addition to newly developed PowerPC software, Apple expects that nearly all of the thousands of current Macintosh software applications will run unaltered on the new systems.

Apple to License Apple DocViewer

"Apple DocViewer is great. How can I build my own documents?"

"Our company wants to distribute high-quality technical documentation electronically."

These are examples of some of the comments Apple has received since it began to publish technical documentation using its electronic viewing tool, Apple DocViewer. Apple has published over 30,000 pages of technical documentation in Apple DocViewer format over the last year.

Apple Computer, Inc., is now making Apple DocViewer available for site license so that you can use it to publish electronic versions of your own documentation. If you license the viewer, you'll also be able to distribute it with your products.

Apple DocViewer provides a powerful, easy-to-use tool for building and viewing electronic documents on the Macintosh computer. Documents published using DocViewer offer users the following advantages:

- Documentation in DocViewer format cannot be altered.
- Users can copy text and graphics from DocViewer files.
- Pictures can be opened at actual size.
- Users can set bookmarks for frequently referenced sections.
- Each document includes a table of contents and index with live links to the page.
- Users can browse through documents using advanced search capabilities.

Apple DocViewer contains two parts—the builder and the viewer. The builder allows you to create Apple DocViewer documents, converting source files into Apple DocViewer files. The builder converts files from any type of word-processing or desktop publishing application, but only files originally created using Microsoft Word or FrameMaker will provide users access to all DocViewer viewing features.

The viewer allows Macintosh computer users to read files built using the Apple DocViewer builder, providing an intuitive interface for searching and browsing documents.

The site license fee is $2500 for version 1.0 of the viewer and builder products. Licensing allows anyone in your company to use the builder to create Apple DocViewer files. It does not provide you with upgrades, however, and Apple provides limited support for the product. Primary support is provided through an AppleLink discussion board (path—Developer Support:Developer Talk:Developer Press:Apple DocViewer) and, for a fee, through third-party support providers (for a list, see the AppleLink location just described).

Distribution agreements granting the rights to distribute only the viewer part of Apple DocViewer to customers are available as well. The fee is $2500 per year for up to 10,000 copies and $5,000 for over 10,000 copies.

For more information and copies of the site license and distribution agreement, send an AppleLink message to SW.LICENSE, or call (408) 974-4667. For more information about Apple DocViewer, send your questions by AppleLink to ADV. ♣

Technology

CD Highlights

System Software Edition October 1993

Hello, and welcome to the October System Software Edition of the Developer CD, featuring over 275 MB of new and revised system software, technical documentation, tools, and utilities.

In response to your feedback about the System Software Edition, this month we inaugurate three new folders in the System Software folder:

• *U.S. System Software:* No longer hidden at the bottom of the Worldwide folder, this folder now contains every version of U.S. system software since version 6.0.5, the oldest recommended system software version for the Macintosh Plus computer. Older versions can be found on the Developer Services Bulletin Board on AppleLink (path—Developer Services:System Software:Macintosh U.S. System Software).

• *Printer Drivers:* This folder contains a reasonably complete list of the most current versions of printer drivers, both for the Macintosh and Windows environments.

• *Other Apple Software:* This folder contains several miscellaneous goodies such as At Ease, Express Modem software, Hardware System Update 2.0, PowerCD software, and much, much more.

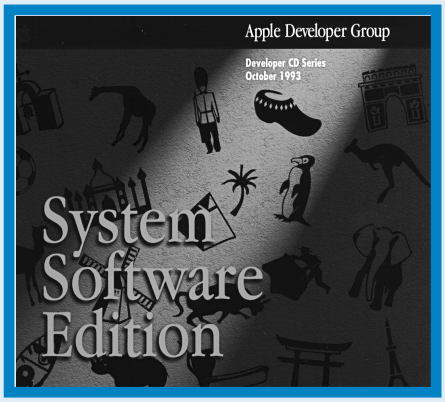
To make room for this new material, we've removed the disk images of localized system software, leaving the more popular network install folders. The Read Me First file in the Worldwide System Software folder describes how to create system software floppy disks from the network install folders.

This month's CD features the following new and revised packages.

System 7 Pro Beta 11

Apple Computer, Inc., will soon release the next addition to the Macintosh System Software product line, System 7 Pro. It includes PowerTalk (AOCE extensions and new Finder), AppleScript, and QuickTime 1.6. The beta 6 version of System 7 Pro was seeded to you previously on the AOCE Beta July 1993 CD.

The purpose of this seed is to allow you to verify your compatibility with the System 7 Pro release and to further your development and use of PowerTalk. Use of the PowerShare



System Software Edition

servers is optional (as PowerTalk is independent of any particular messaging or catalog service), and we recommend that you determine if your work requires the use of PowerShare servers before you install them.

Included on this disc (in the System 7 Pro Beta 11:Development Tools:Testing Materials folder) are two documents that provide suggested testing scripts for verifying your product's compatibility and savviness, respectively. These documents are by no means all-inclusive, but we recommend them as a starting point.

CPU System Enablers

This folder contains system enablers for several Macintosh models. This month features an updated version of System Enabler 131, which replaces enablers 111 and 121 and also supports the new PowerBook 180c computer; System Enabler 088, for the Macintosh Centris AV and Macintosh Quadra AV; and System Enabler 408, for the Macintosh LC 520.

Disk First Aid 7.2

Disk First Aid 7.2 contains a new interface and fixes a much wider class of problems than before. It includes all of the corrections in Disk First Aid 7.1.1.

Macintosh Human Interface Guidelines

This is the Apple DocViewer version of the *Macintosh Human Interface Guidelines* book from Apple's Developer Press. The printed version is currently available through Addison-Wesley. The on-line version features all the functionality that Apple DocViewer provides. Screens and art are in 8-bit color.

MacCheck

Apple designed the MacCheck program to provide configuration information and software problem detection for the use of Apple phone support people and knowledgeable consumers. It's intended for systems

Please turn to page 13

Inside This Section

Human Interface

I've Seen the Future

By Peter Bickford

I'm writing this article while on sabbatical. To my thinking, an employee sabbatical has to be just about the greatest, most innovative, and ultimately humane policy at Apple. The deal is that after every five years of service, Apple employees are given several weeks off to kick back, get their heads together, go backpacking in Greenland, or whatever. Given the work schedules of some of our more dedicated programmers, sabbaticals may be their big chance to discover that bright yellow orb most of us know as the sun.

For a few blessed weeks, at least, a sabbatical means time to reflect. Of course, this applies to those employees who don't have a mean, terrible ogre of an editor who—sabbatical be damned—insists on getting an interface column out of them every month. It's probably just as well, though, because I've noticed that after an appropriate amount of reflection, a surprisingly large percentage of sabbatical-takers decide to pitch it all and write travel books, or become tanning instructors in Fiji. Of course, I wouldn't want to take the chance of a similar tragic fate befalling me. Instead, I decided to use the time to ponder the evolution of human interface, and to try to distill a few drops of wisdom about the direction it will take in the future.

That is, I spent my sabbatical playing video games.

State of the Art

If you want to keep up with the cutting edge of computer technology, you can take the standard approach. This involves visiting as many advanced research labs as possible, reading lots of dry technical journal articles, and trying to make friends with people who hold high government security clearances.

If any of these ideas repulses you, you'll be glad to know that there's another way to go: Simply get a large roll of quarters and head down to your local video arcade.

Color screens, high resolution video, reactive input devices, voice capability—the games define the state of the art. While the rest of the world was busy getting syntax errors using command-based interfaces, the video game world was completely GUI-based. Years before the trackball was saving computer users from cluttered desks, it was helping video game players save the world in Missile Command. And scarcely had the multithousand-dollar neuron glove been featured in *Scientific American* than Nintendo players were using a $39 version to fend off the bad guys.

Recently, a new breed of entertainment has appeared that's radically changing what people expect from video games. It also tells us what people will be expecting from their computers in the future.

I've Seen the Future . . .

Three years ago, an inventor held a private session for a group at Apple that I was part of. In a dimly lit conference room, he showed us a couple of breadboards covered with an assortment of loose circuitry and wires. He then hooked up some of the wires to a specially engineered tape deck and other wires to a couple of good speakers. Then he made a perfunctory speech, turned down the lights, and blew us all away.

The inventor had created something very simple: a way to receive several streams of video at once, and have a simple computer switch between them in accordance with a simple program. That's all very well and good, but as technology goes, it wasn't exactly the space shuttle. What made this man a genius was not the technology he'd invented; it was the application. Instead of some dull industrial application, this inventor used his switching technology to create a whole new generation of video games.

Blessed with sufficient capital, the inventor had hired known actors to star in his software, and enlisted none other than Industrial Light and Magic to do his set design and special effects. The "interactive movies" he demonstrated included a murder mystery in which you could follow the action from room to room as the story played itself out in real time, spying on different characters from surveillance cameras located throughout the house. When the murder had been committed, the suspects all lined up outside, and you had to point to the one who had done the dirty deed. If the maid was guilty, but you picked the old reliable butler, he would step forward on video to defend himself—perhaps protesting that you knew he couldn't have done it because you had seen him on the other side of the house only moments before the crime had been committed.

Impressive as this was, the real showstopper was called Sewershark. The setting for that game was a futuristic fighter base located in the sewers below a city. In that grimy, steam-filled environment, you took on the role of a rookie pilot. Actress Maria Conchito Alonzo plays the part of a hard-bitten veteran who gives you your initial briefing. Speaking directly to the camera, she informs you that your job is to fly your ship down the tunnels at blazing speeds, shooting down various nasties and trying like heck to avoid crashing into a wall. Careful observation of her attitude suggests that she doesn't think much of your chances of survival.

Your flight instructor is even more nasty, periodically busting in on your view screen to give you evaluations of your performance. If you do badly, he tells you—in no uncertain terms! If you do well, he expresses amazement that you haven't messed up too badly yet. Or so we hear; none of the people trying out the demo got to enjoy that message. We did discover, however, that the technology provided the flight instructor with a large and varied list of put-downs for use in grading our performance.

After a few goes at what basically amounted to whooshing through a 70-mm movie, complete with state of the art sound and visual effects, we reluctantly surrendered our joysticks and let the inventor pack up and head home.

Around Christmas of this year, Sewershark appeared as a commercially available product. The video resolution isn't as good, but otherwise, it's identical to that amazing demonstration I saw three years

earlier. The game costs about $49 at your local K-Mart and runs on a Sega Genesis home video game system.

Back at Work—The Shape of Things to Come

While "serious" interface designers argued whether color really added anything to an interface, legions of Space Invaders fans already knew the answer. If you ever doubted that sound can have a major impact on performance, just try playing Defender with the sound turned off. And if you're skeptical about the possibilities of multimedia, I suggest you pick up a copy of Sewershark.

Video game designers have a lesson to teach us mainstream interface designers—and it's about time we listened. It's silly to act as if our users are deaf, are colorblind, have limited range of motion, and have no sense of touch. Instead, our technology needs to use as much of the human experience as possible.

The future is polyphonic, three-dimensional, high resolution, full color, full motion, and touch-sensitive. In other words—like real life.

So, my advice for all of you designers out there is this: Dream a little. Go beyond the dull expectations of your application's genre. For example, if you write utility programs, look into using animations for status indicators. (I've been told that one well-known utilities package employs a running man that's one of the most popular features of the program.)

And, if you're lucky enough to be a game designer, keep pushing the envelope. The rest of us will be right behind you!

Till next time,
Doc
AppleLink: THE.DOKTOR

Having survived his sabbatical leave, Pete Bickford is still a member of the Apple Enterprise Systems—now called Apple Business Systems—human interface team.

Rethinking Your Applications for QuickDraw GX

By Kris Newby

For end-users, the incorporation of QuickDraw GX imaging technology into applications promises to be as dramatic as when Dorothy, in the movie *The Wizard of Oz,* stepped from her black-and-white Kansas farm into the Technicolor land of Oz. Over-the-rainbow color, sophisticated graphics, and quality printing will soon be everywhere—not just in expensive, high-end products.

If you're the great and powerful graphics wizard of your company, you'll find that QuickDraw GX makes it easier for you to create the visually rich applications that customers are demanding. And even if your applications aren't graphics-intensive, you may be surprised at how easy it is to add sales-generating features such as high-quality output, easier formatting, and document portability.

From a marketing perspective, QuickDraw GX really cements Apple's leadership position as the best mainstream graphics computing platform, moving it years ahead of competitors. You can leverage this position and new technology to gain a competitive advantage in your own market. By incorporating QuickDraw GX features into an existing product, you can give your current customers more reasons to upgrade. By using new capabilities such as dynamic type manipulation, multilingual text support, and device-independent color matching, you can create breakthrough products and enter new markets.

In this article we'll tell you why it's important to map out your transition to the QuickDraw GX architecture now and give you some ideas on hot features you can add to your products. Then, we'll offer performance-enhancing tips on working with QuickDraw GX calls, objects, and memory.

A Quick Overview of QuickDraw GX

QuickDraw GX is an API of graphics, typography, and printing routines that developers can quickly access to perform complex imaging tasks within their programs. QuickDraw GX will ship as an easy-to-install system software extension that augments the capabilities of the original QuickDraw—a set of graphics routines built into Macintosh Read-Only Memory (ROM). This new API consists of three parts that run on top of System 7.1, as shown in the schematic below.

QuickDraw GX promises to make any development work that involves printing, type, or graphics faster and more interesting. Access to built-in routines will result in less code that you'll have to write, debug, and maintain over the life of a product. And this means you'll be able to spend more time working on the fun stuff—creating new features, products, and user interfaces.

Getting Savvy to GX Product Enhancements

Since QuickDraw GX coexists happily with the original QuickDraw, users will be able to run existing applications "as is" when QuickDraw GX begins shipping later this year. Users benefit from a number of "out-of-box" features such as drag-and-drop printing to multiple desktop printer icons, font embedding, and document portability.

By making just a few code modifications, you can make sure your applications are well-behaved and "GX-aware" under this extension. GX-aware applications need to support a new printing interface, portable digital documents (PDDs), and a new GX Clipboard.

By wading in just a little deeper, you can make your applications "GX-savvy"—in other words, you can set new standards in the use of graphics, type, printing, and color. (For detailed guidelines on becoming "GX-savvy," see the

August '93 Developer CD, *CD Slickers*, or "Getting Started With QuickDraw GX" in *develop* Issue 15, September 1993.) There are three strong reasons for investing time and energy to make your products GX-savvy.

First, it's a great opportunity to differentiate an existing product by adding meaty QuickDraw GX features to an upcoming revision. With relative ease you'll be able to add completely new capabilities, or ones that were previously available only from high-end competitors. (See the chart, "QuickDraw GX–Based Feature Ideas" on page 12 for hot ideas on type, graphics, color, and printing functionality that you can add to your specific products.)

Along those same lines, you can localize an existing product for untapped worldwide markets by taking advantage of QuickDraw GX's international type enhancements. With QuickDraw GX and WorldScript combined, you don't need to create custom support for complex text systems such as Arabic, Hindi, or Japanese Kanji. At the system level you get support for two-byte characters (for non-Roman languages), vertical-flow text, and automatic contextual letter form substitutions (for Hindi and Arabic). And since QuickDraw GX lets you manipulate type like any other object, you'll also be able to rotate, skew, and add perspective to any of these text systems.

Larry Oppenberg, VP of Type Operations at Bitstream, thinks QuickDraw GX will breathe new life into the font market. "We see QuickDraw GX starting a renaissance in type. Not only will it help us innovate and add value to our existing type products, but it opens worldwide markets to us. For example, we think the Asian market is on the verge of explosive growth, since localization issues have limited the available fonts in these countries."

Second, QuickDraw GX gives you the tools to create innovative new products, so you can expand your market presence. Here are just a few GX-based product ideas:

- randomized, handwriting-like font families, using advanced functions like ligature substitution and automatic kerning
- a printing extension that helps service bureaus track and bill for use of expensive output devices
- a printing extension that helps graphic designers preview artwork for color, bleeds, mixing, screens, and trapping before expensive color-separations are made
- inexpensive, easy-to-use image editing for use with the growing number of color input devices
- a PDD viewer that provides editing tools for reviewing all types of PDD files, from brochures to presentations

Third and last, by learning QuickDraw GX now, you'll get a jump on the transition to the PowerPC processor–based Macintosh. And you'll be in line with Apple's future system development directions. QuickDraw GX will reside on the PowerPC Macintosh in native mode at this product's introduction.

Performance Tips From Our GX Wizards

Overall, using QuickDraw GX's sophisticated set of system-level graphics routines will enable you to provide richer features to your users in less time. And your GX-savvy applications will be much smaller than comparable applications today, requiring considerably less hard-disk space. But to get the best possible performance out of the new QuickDraw GX engine, you'll have to fundamentally change the way you work with "shapes"—the basic building block of QuickDraw GX graphics.

The biggest difference between QuickDraw GX and the original QuickDraw is that shapes are built under an object-based rather than a state-based paradigm. A shape is an object that contains, among other things, a geometry and a fill property that specifies how it should be drawn. All of a shape's attributes—things like size, color, fill, perspective, and line thickness—travel with that shape, rather than existing as modifications to the grafPort's previous state. This self-containment makes it easier and faster for your programs to manipulate and export shapes.

You'll also have to be more careful about how your QuickDraw GX–based application uses memory, since performance is directly related to how much memory is available in a user's system. We recommend that you keep the footprint of your application as small as possible, since QuickDraw GX intelligently grabs spare memory to speed up processing time.

To help you get up and running with QuickDraw GX, we asked Apple's own QuickDraw GX wizards to put together some specific performance-enhancing tips. Though they're currently busy debugging the beta version, they imparted a few gems of wisdom to us while on their way

QuickDraw GX Developer Benefits

Here are five main categories of QuickDraw GX capabilities that will make graphics-related development faster and more interesting.

- Built-in system-level imaging routines allow you to create applications with a leaner and easier-to-maintain code base. Rather than providing a set of drawing commands, this system is built around a graphics database that provides an extensive set of geometric elements, shapes, styles, and transformations that applications can draw upon as needed.
- Enhanced system-level support for typography, graphics, and color control enables you to enhance existing applications and create new types of products. New capabilities such as object-oriented type manipulation, multilingual text support, and device-independent color matching give developers the tools to create breakthrough applications.
- Device-independent display and output hooks make it faster and easier to support a wide range of peripheral devices. QuickDraw GX is based on mathematical curves and outlines, so that applications don't have to be dependent on specific display or output resolutions.
- A new print architecture streamlines driver development, potentially reducing the code required for specific drivers by as much as 95 percent. This architecture is also open, making it possible for developers to create custom printer driver extensions.
- Uniform page-definition guidelines make it easier to transfer formatted information between applications. Documents saved in the new QuickDraw GX independent file format (Portable Digital Documents, or PDDs) can be opened, viewed, and printed on any Macintosh system that has QuickDraw GX—without the original application or typefaces. ♣

to the caffeine machine. Eleven of these useful tips follow:

• *Reuse shapes when possible.* Once you build a shape using QuickDraw GX, the system is extremely fast at redrawing it over and over again under the same conditions. For faster overall application performance, reuse shapes and redraw them as often as possible. This strategy also works for view ports, when you need to implement scrolling within multiple windows, views, or panes.

• *Store shapes wisely*. The graphics and layout portions of QuickDraw GX use their own memory heap format and Memory Manager to improve efficiency. There are two ways you can save a shape created with QuickDraw GX: You can "store" it as is, making it readily accessible in the GX memory heap, or "flatten" it as a compressed data stream on disk. Shape manipulation is obviously much faster when shapes are stored in the heap, but if you store everything, application performance will bog down. This is because QuickDraw GX pages shapes out to disk memory when RAM becomes tight. This wasted operation slows your application down. The best approach is to adopt a storage strategy that anticipates how users will work with your application. For example, with multipage documents, have your program save all the shapes on the current page, and possibly the page before and after. Store shapes that are drawn repeatedly or in response to an update. Discard shapes as soon as you're through with them.

• *Use font variations.* Take a hierarchical approach when working with fonts, in order to reduce font family memory requirements and make it easier for users to embed fonts into documents. Since descriptive font information travels with QuickDraw GX fonts, you can create a "styled" version—such as bold, condensed, or light—by specifying a variation of the original outline font.

• *Don't split pixels in gradient fills.* Use single pixel-wide bitmaps to describe linear ramps used for filling shapes with color gradients. This strategy prevents rounding-error "jaggies" when you rotate, scale, and clip a shape filled with a ramp.

• *Group little shapes into bigger ones.* Performing operations on lots of little shapes is almost always slower than working with larger shapes. So group curves and lines into paths and polygons where possible. For example, when drawing an on-screen ruler, group the tick marks into a larger shape, such as a polygon. And along the same line of logic, keep users from creating too many closely spaced dashed or patterned elements; the resulting shape may eat up memory and slow down performance.

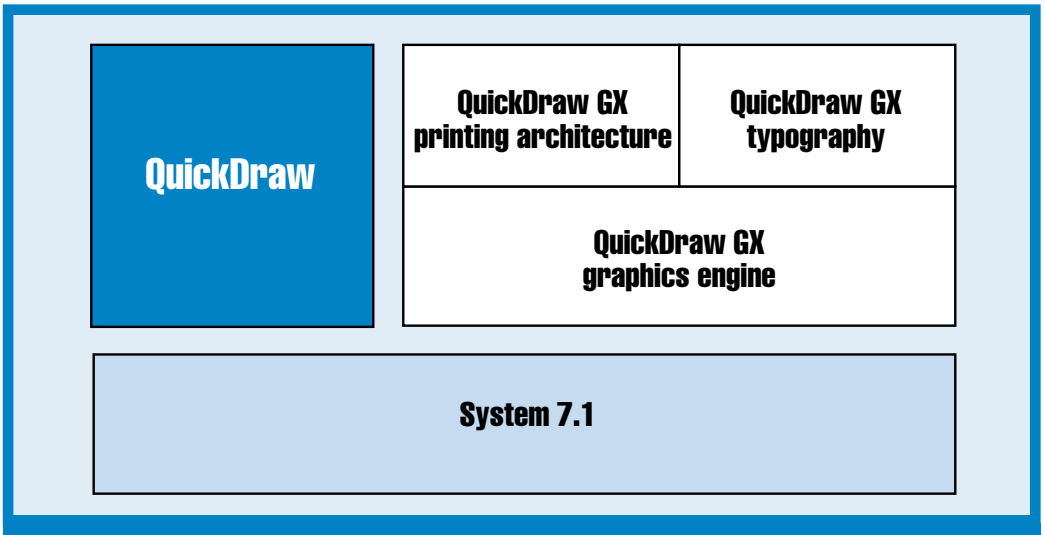
• *Use GraphicsBug to look under the hood.* The only way to create and modify shapes in QuickDraw GX is through the public API; you can't operate on data within the shape directly. This approach lets Apple expand the system in the future with minimal compatibility risk—but makes it difficult for you to debug your program. That's where GraphicsBug comes to the rescue. This tool lets you "look under the hood" at the contents of any QuickDraw GX object to make sure it contains correct information. In addition, the debugging version of QuickDraw GX provides extensive validation facilities that determine whether you're passing valid data to the QuickDraw GX API and whether anything's gone awry internally.

• *Minimize fragmentation of the GX memory heap.* Avoid locking up large blocks of memory with calls such as LockShape or LockTag. When you have to use these calls—for example, if you have to manipulate bits within a bitmapped object—unlock this memory as quickly as possible. If you fragment GX memory too much, you increase the likelihood that calls will restart.

• *Don't hard-code custom colors.* Define colors within files in their native format, using their original colorSpace and colorProfile values. This ensures that exported graphics data carry the maximum precision for use by other applications. And when possible, define color mixes (transfer modes) so they don't

Additional QuickDraw GX Resources

- "Getting Started With QuickDraw GX," *develop* Issue 15, September 1993
- "Developing QuickDraw GX Printing Extensions," *develop* Issue 15, September 1993
- "QuickDraw GX for PostScript Programmers," *develop* Issue 15, September 1993
- "Looking Ahead to Printing With QuickDraw GX," *develop* Issue 13, March 1993
- "What Is QuickDraw GX-Savvy?" and "QuickDraw GX Product Ideas," *CD Slickers* Developer CD or the WWDC '93 CD
- The white paper "QuickDraw GX: The Color Publishing Platform" on the WWDC '93 CD



QuickDraw GX architecture.

QuickDraw GX–Based Feature Ideas

QuickDraw GX imaging technology offers you a wide variety of built-in resources to help you differentiate an existing product or create new ones. This chart lists just a few ideas for innovative product features that are relatively easy to add using this powerful system extension. Look up your own application type in the left-hand column to browse through some GX-based feature ideas that you'll soon be able to implement.

	NEW GX FUNCTIONALITY				
APPLICATION TYPE	**Typography**	**Graphics**	**Color**	**Printing**	**Portable digital documents (PDDs)**
Word processor	Better type quality automatically, e.g., kerning, tracking, and automatic fractions	GX Clipboard support for importing multiformat graphics	Predictable, easy-to-use color graphics and text	Controls for by-page formatting, e.g., multiple page sizes and orientations saved in one document	Embedded fonts for document printing and viewing by all, even without original fonts resident
Page layout	Advanced, easy-to-use type features such as ligatures, dynamic manipulation, and style variations	High-end graphics and type tools for masking, rotation and perspective effects	Color matching and separation tools for high-end publishing	Controls for background screens and by-page formatting; prepress previews of traps, bleeds, etc.	Easy review of complex documents and output to service bureaus without original fonts or applications resident
Graphics	Sophisticated graphical manipulation of resolution-indendent, editable type	Light-source shading of illustrations and real-time color mixing	Accurate color blending and user-selectable color management	Customization of the new color picker for using industry standard color systems	Easy review of embedded illustrations without original fonts or applications resident
Presentations	Dynamic, resolution-independent type manipulation	Built-in high-end graphic creation and editing tools	Consistent color across paper, foil, and monitor slides	Audience note printing without graphics or background drop-outs	Reliable presentations that work on the road without original fonts and graphics

depend on the pixel value in the view device. This will help performance and make it easier to do color matching.

• *Avoid unnecessary use of floating point macros.* Don't use the fl macro (floating-point to fixed-point number) in operations when the ff macro (integer to fixed-point number) will do. The fl macro introduces both run-time and compile-size overhead that could slow down your application.

• *Use fast code paths where possible.* Use shape editing routines instead of creating a new shape to change an esixting shape, and use QuickDraw GX calls like TouchesShape or ContainsShape rather than IntersectShape or DifferenceShape. And use SetShapeParts (or the library call AddToShape) to accumulate a series of shapes, and then a SimplifyShape call on the result, rather than repeatedly calling UnionShape. These functions require less time-consuming calculations.

• *Avoid unnecessarily turning on color matching.* Since color matching is memory intensive, avoid turning it on when it's not needed. One memory-saving strategy for 8-bit screens is to create a user color set that contains only matched colors, so users don't affect performance by using colors they can't view or print.

Our Success Formula: Brains, a Heart, Courage, and QuickDraw GX

QuickDraw GX really cements Apple's leadership position as the best mainstream graphics computing platform, and you can leverage this position to gain an advantage in your market. By incorporating QuickDraw GX features into an existing product, you can give your installed base more reasons to upgrade. By using QuickDraw GX to develop a new market niche, you can expand your company's market share.

From a developer's perspective, QuickDraw GX enables you to concentrate on creating breakthrough product features rather than reinventing, maintaining, and adding to a large graphics engine. And this ultimately frees you to do what you do best—innovate.

We hope the advice in this article gives you ideas and saves you development time, but we subscribe to the Wizard of Oz's advice on being successful—that you already have the brains, heart, and courage within to create great products. We just provide helpful tools. ♣

Kris Newby, principal of Kris Newby Technical Communications, is a marketing communications consultant and freelance writer based in Palo Alto, CA.

	Typography	Graphics	Color	Printing	Portable digital documents (PDDs)
Spreadsheet	Improved page formatting, e.g., auto-fractions and columnar data alignment	Tools for creating perspective, rotated, and animated charts	Consistent color across paper, foil, and monitor slides	By-page formatting allowing multiple page size charts in one document	Easy review of complex documents by all, without fonts or applications resident
Database	Easier, higher quality formatting of reports	Import and output of all GX graphics types (bitmap, vector, etc.) without quality loss	Tools for creating image databases, e.g., an employee photo and info database	Creation of usage-tracking printer extensions for chargebacks of output device use	Easy routing of database reports for review
CAD	Object-based type for projection and manipulation on surfaces and curves	Resolution-independent graphics for more accurate perspective and rotation	Color blending and transparency for clearer 3-D diagrams	Support for a large range of plotters	Easy review of complex documents by all, without expensive applications resident
Multimedia and animation	Conversion of sprites into dynamic, animated fonts	Easier image manipulation and display with resolution-independent graphics	More accurate color mixing, and object interaction for presentation on any size display	Creation of a storyboard application for accurately previewing animations and color with clients	Distribution of a storyboard application on line for client review
Output devices	Better output quality automatically with resolution-independent type	Non-PostScript device printing of previously PostScript-only graphics	Accurate color printing across multiple devices	More control of hardware, e.g., creation of toner and paper bin controls	Automatic billing of print processing time triggered by scripts
Fonts	Automatic font styles based on text location and kerning info built into type	Font manipulation tools for creation of custom glyphs, icons or logos	Intelligent type that automatically changes color according to background color	Resolution-independent, high-quality output on any device	Creation of embeddable fonts that travel with documents for reviews, etc.

CD Highlights

continued from page 7

running System 7.1, although it will run on all systems using System 7 and later versions. It provides configuration information, Disk First Aid (7.2) verification code, system file corruption detection, duplicate System Folder detection, and a logic board test.

New Memory Manager

This new Memory Manager uses better data structures for performance. It also includes a lot of debugging checks to catch illegal operations on the Memory Manager (such as disposing of a fake handle). The CD contains a system file that runs on most 32-bit clean machines. This system file runs the new Memory Manager in client debug mode. Also included is a Memory Manager–friendly version of MacsBug and debugger preferences. Developers should test programs with this system to ensure Memory Manager compatibility in the future.

PowerPC Developer White Paper

This 28-page white paper provides an overview of the upcoming PowerPC processor–based Macintosh platform specifically for third-party Apple developers.

QuickTime 1.6.1

This version of the QuickTime system software extension adds capabilities that let your programs integrate graphics, sound, video, and animation into documents. By providing a standard way for all Macintosh programs to control these multimedia elements, QuickTime makes them easier to use.

Here are some of the new features of QuickTime 6.1:

- smaller size
- ability to convert audio tracks to movies
- Macintosh Easy Open support
- ColorSync support
- improved grayscale
- smoother text
- Cinepak compression-decompression software

Next month's CD will provide several long-promised yet elusive goodies, such as the Drag Manager, a new QuickDraw GX beta version, a network game demo, and more. See you there!

Alex Dosher
Acting Developer CD Leader

Ask Don Norman: Conversations on Technology and Society

Will Tomorrow's Computer Be Able to Read Today's Files?

By Don Norman

I recently received the following note on AppleLink from a developer and journalist in Sweden:

Thanks to the revolution in electronics, we are now creating huge databases (on CD-ROM discs or networks), and a new wave of hyper literature (facts and fiction) is at hand. A point stressed by so many is that this distribution of information is the beginning of a new democratic era.

But what happens in 100 years? If the James Joyce of the 21st century publishes a multimedia novel with the help of a certain authoring tool, a certain version of QuickTime, and a certain version of System 7, will a computer be able to read it in the year 2093?

Storing on magnetic and/or optical media saves much space compared to storing piles of paper and rows of books, we are told. But what if the libraries of the future have to be museums too—with hundreds of antique computers set up in hundreds of different configurations?

Someone at Apple here in Sweden said to me, "There will always be people who can decipher old files—if we can interpret the Dead Sea Scrolls today, computer files will not be any problem." Some future hacker with an inclination toward cyber-archaeology will gladly spend weeks reconstructing old Macintosh files.

But what about easy access for everybody? What about making the unlimited knowledge of the world only a mouse click away? (This was a vision H.G. Wells wrote about as early as the 1930s!!!)

I know there is a lot being done today when it comes to standards between different computer platforms, languages, and so on. But does anyone think of standardization over time, over decades, centuries?

Isn't it ironic that creators of more short-lived material, such as presentations, can use the latest technology, but anyone who produces literary or educational material meant to last some decades, has to limit himself/herself to some sort of lowest common denominator when it comes to technological complexity?

Shouldn't we set up at least some guidelines for minimizing the risk that, for example, future font resources will be completely inadequate for showing our old texts to the future, even in a jagged fashion? How can we anticipate compatibility problems created by future ultra-fast processors?

Dare we believe that libraries in the future will be equipped with super-computers that can emulate or simulate various sorts of antique hardware and software? Shouldn't this emulation or simulation process of the future start here and now—with discussions, plans, and guidelines for developers of authoring tools, the invention of special "embedded reconstructional code sets," and who knows what else?

Regards,
Karl-Erik Tallmo
Nisus Publishers
Stockholm, Sweden

Thanks for the interesting commentary and question. The problems you raise are indeed serious. I discussed some of these issues in a lighthearted way in my essay "I Go to a Sixth Grade Play" in my book *Turn Signals Are the Facial Expressions of Automobiles,* part of which is reprinted here (see side bar).

I don't think there are any simple answers to your questions. The problem occurs with any technology that represents information through what I call "internal representation" rather than through "surface representation," both of which are explained in the excerpt from my book.

The problem exists not only for all computer records, but also for lots of things that do not deal with computers at all—for example, audio recordings on wire spools or 3/4 inch tape at peculiar speeds; video recordings on a wide variety of formats; photographs stored on CD-ROM discs (Photo CD) or with digital cameras; and insurance information on punched cards (or, for that matter, anything on punched cards). We have already lost lots of data for this reason, as the excerpt shows.

I know people with lots of data on punched cards, punched paper tape, and "DEC-tape" (who remembers that?) that is now unreadable for the same reasons. People filming their families with video cameras today may discover that their children and grandchildren will not be able to view the scenes because, by then, the technology will be very very different. (Who today could play back an audio wire recording? Very few people have even heard of the technology; fewer still have access to machines that can work with it.)

The adoption of standards will not help, because this problem goes beyond standards: Standards work well within a technology, but this problem occurs because new technologies are steadily introduced, making the old obsolete. Thus, we now have some standards for recording video, but within the next decade the world will completely switch video formats from analog to digital and, most likely, change from today's mix of PAL, SECAM, and NTSC standards to one worldwide HDTV standard. A decade or two after the switch, we will likely be unable to read information recorded prior to the switch.

I asked Steve Cisler of the Apple Library, who's working on the Apple Library of Tomorrow grant program, for his opinion on your problem. He had the following to say:

The idea that you can always move digital formats is quite true, assuming you can start the old machine and move the data

to the new model, but I don't know of any large institution that has done a massive data translation from, say, an old set of tapes to some new format. This goes for the U.S. Census and the National Center for Supercomputer Applications.

One of the problems is that the market for archival preservation is not very large, except for financial records, so few companies are going to get into the business.

I think most of the projects will be recovery oriented, rather than performing routine conversion as a maintenance operation, so a lot of less important information may not be moved to new formats.

From time to time people will come into the Apple Library Lab, and they'll want to use an Apple II+ or a Lisa or 128K Macintosh; we have them, but Apple has only been around 15 years or so. Will the company support this for 150 years? Imagine the space needed for just our equipment and you can see the challenge posed by the technology, the competition, and the differentiation in products if an organization is trying to archive electronic materials from many sources.

If you're ever visiting Silicon Valley, take a trip to Ampex in Redwood City; at their peak they were the top audio/video recording company. They have an excellent museum (where you can see wire recorders "liberated" from German labs) and the early recording units funded by Bing Crosby who was a major shareholder.

I think the problem you describe is real and serious. Alas, I know of no simple answers. I don't even know of any complex ones. I recommend, even in this age of digital media, that we keep important material archived on paper or movie film. As you say, isn't it ironic that those of us in the forefront of technology must limit ourselves to the lowest common denominator of technology? ♣

Don Norman is one of three Apple Fellows (Don, Alan Kay, and Gary Starkweather—the jolly good fellows, they call themselves). He holds a few other titles, as well: User Experience Architect for Apple, Professor Emeritus, University of California at San Diego, and founding chair of the Department of Cognitive Science there. He's also the author of lots of books, the latest being Things That Make Us Smart *(Addison-Wesley, 1993). Send your questions to him on AppleLink care of A.DIRECTIONS; he'll answer them as his busy schedule allows, and we'll publish the answers here on a semiregular basis.*

From *I Go To a Sixth Grade Play*

In my studies of technology, I have distinguished between two different modes of representing information. An artifact can have a surface representation or an internal one. A *surface* representation is like the words on the page of this book. The information is all there on the surface: what you see is what there is. Photographs, drawings, letters, and books all exploit surface representations.

The advantage of surface representation is that no technology is needed to experience it: all one has to do is look, and there it is. All that matters is that the artifact still exists, and the marks on the surface are still visible.

This is not as simple as it might seem. In the past hundred years, most paper has been manufactured using an acid-based technology that destroys the paper itself after some time. Most of us have experienced the deterioration of newspapers: leave one exposed to the light, and a year or two later, it crumbles and fades away. Books and personal notes have a similar, if slower deterioration. Today, paper manufacturers recognize this problem and most quality books are published on acid-free papers. But notepads are not.

Photographs also fade. Slides, color prints, and black and white photographs last only as long as their chemically activated marks stay there, as long as the paper on which they are printed is strong. Prints last longer if you don't expose them to the light. So photographs last longer if you don't look at them. Fortunately, we can expect books, notes, and photographs to last over 100 years, long enough for most purposes, certainly longer than the lifespan of the individuals who made them.

The story is quite different for those who entrust their records to *internal* artifacts, artifacts where the information is stored invisibly inside the material, where technology is required to retrieve it.

Remember the days when we recorded voices on spools of wire, rapidly passing before a magnetic head? You probably don't remember the forerunner of today's tape recorder. But that is just the point. If I recorded my baby's first words on a wire recorder, or perhaps had the sole surviving recording of President Roosevelt's farewell words to Joseph Stalin and Winston Churchill, they would be virtually unrecoverable. Unless someone could discover a working version of the wire recorder (or could make one), there would be no way to recover those sounds again. Or suppose I had recorded the sounds on one of Edison's wax cylinder phonograph records.

Do you think the videotape technology we use today will be around fifty years from now? No way. Today's videotapes will not be playable. Nor audio tapes. Nor phonograph records.

The same, sad story is true of all technologies that use internal storage: that is, any recording device that requires another technology to reproduce it.

The same problems have already affected science. The American National Aeronautics and Space Administration (NASA) has tens of thousands of reels of magnetic tape, carefully preserved, with data recorded from space and satellites. From those data we could follow in detail the earth's weather patterns and cycles of plant growth, or water, or pollution. Except that they are recorded on an old-fashioned tape with an old-fashioned format that can no longer be read by today's computers.

I have not even discussed the fact that the magnetic particles on computer, video, and audio tapes and wire recordings do not last forever. They slowly migrate, losing their signals. And the tape itself cracks and deteriorates.

Nothing in this world is permanent, nothing lasts forever. But most of us had not counted on technology's assistance in hastening the demise of our memories.♣

Business & Marketing

Market Research Monthly

System 7 Installed Base Global Analysis, Part II

Inside This Section

Last month's Market Research Monthly featured the first release of Apple's proprietary System 7 penetration data in the United States. (See "Exclusive: Apple System 7 U.S. Installed Base Numbers" on page 27 of the September issue of *Apple Directions*.) This month we bring you the rest of the analysis, as promised: global System 7 installed base data.

As in last month's article, we provide these numbers to help you decide whether your applications should support only System 7 or both System 6 and System 7. As you'll see, you may need to make a different decision based on the part of the world in which you market products.

The figures come from a telephone survey Apple Computer, Inc., conducted in April and May of this year with nearly 4000 Macintosh computer users who returned their registration cards between 1988 and 1993. The users come from nine key countries representing Apple's three geographic sales regions, the United States, Europe, and Pacific. European countries covered by the study are the United Kingdom, France, Germany, Italy, and Sweden; so-called Pacific countries are Japan, Australia, and Canada.

System 7 Adoption Rate

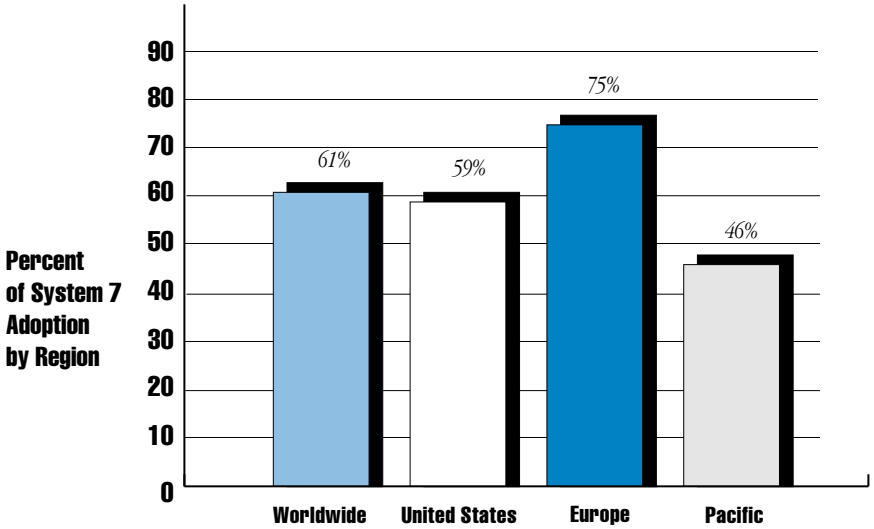
The study determined that, overall, System 7 has been adopted by 61 percent of Macintosh users worldwide (see the graph "Global System 7 Adoption Rate, 1993"). Adoption varies significantly from country to country: System 7 has attained the highest adoption rate by Macintosh users in Germany—92 percent—which has a greater proportion of high-end users than most countries. Sweden ranks second with 79 percent of Macintosh users running System 7 (see the graph "System 7 Adoption by Country, 1993"). System 7 acceptance among U.S. Macintosh users reached 59 percent at the time of the global study. (The U.S. figures reported last month were taken from an earlier study.)

Japan has the lowest System 7 adoption rate, at only 28 percent. This is because the two-byte version of System 7, KanjiTalk 7, only began shipping late in 1992. Given that Japan is currently one of the hottest markets for Macintosh sales and that KanjiTalk 7 currently ships standard with every Macintosh computer sold in Japan, this number is expected to rise dramatically in the near future.

Similar to the U.S. pattern, System 7 adoption rates also vary according to Macintosh model worldwide. Users of high-end systems are far more likely to run System 7 than are users of low-end and compact systems. At the high end, 80 percent of Macintosh IIfx users run System 7, while all Macintosh Quadra users run System 7 because their computers ship standard with it. The majority of mid-range users have also adopted System 7; for example, 80 percent of Macintosh IIci users run System 7. Among users of low-end and compact computers, 11 percent of Macintosh Plus users have upgraded to System 7, while just over one-third of Macintosh SE and SE/30 and Macintosh Classic owners use System 7.

Widespread Demand for Color

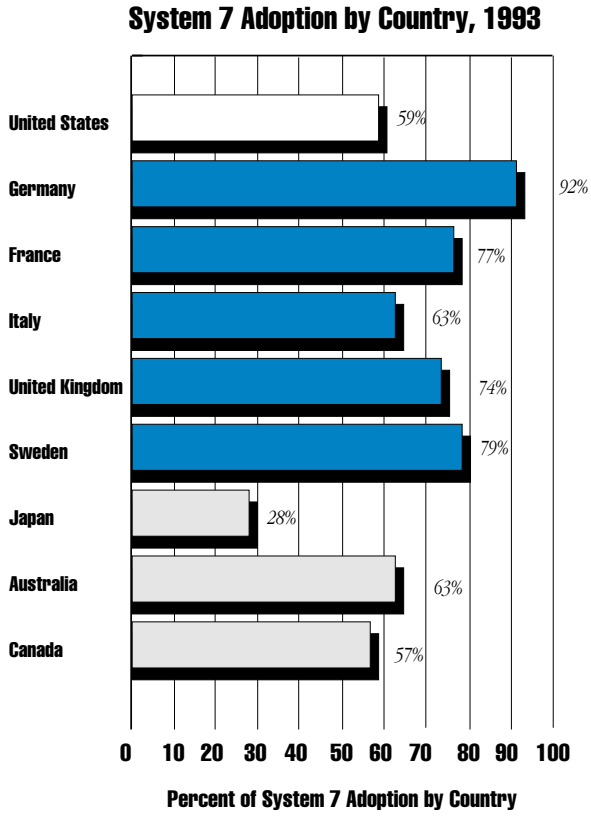
The study also revealed other interesting data about the Macintosh installed base,



which we'll report in another issue. One timely bit of information from the study, given Apple's imminent release of QuickDraw GX (see "Rethinking Your Applications for QuickDraw GX" on page 9), is that Macintosh users around the world are demanding color capabilities.

Worldwide, 43 percent of all Macintosh customers currently use a color display while 81 percent of owners of systems with external displays (as opposed to all-in-one systems with internal monitors, like the Macintosh Classic) use a color display. Apple expects the use of color to become even more prevalent over time, giving you incentive to support the new capabilities made possible by QuickDraw GX.

Let us know what segments of the Macintosh market you need to learn about for help with your business planning. You can send your ideas to us on AppleLink at A.DIRECTIONS. ♣



Marketing Feature

Corporate Partnering

How a Small Developer Can Find and Approach a Larger Company Partner

By Jacqueline Daunt, Fenwick & West, and George Von Gebr, Von Gebr & Tan

"Corporate partnering" and "strategic alliances" are the terms used to describe a symbiotic long-term relationship between two companies for developing and exploiting technology, products, and markets. The nature of the partnering relationship depends on the partners' strategic objectives. It can be as simple as a long-term distribution agreement or an equity investment. More complex arrangements may include research and development contracts, or manufacturing and licensing rights to the resulting technology or products. The key to a successful corporate partnering deal is combining the right set of options, so that the needs of both companies are met.

This article will describe the process of finding partners and outline the steps you could take to begin the partnering process.

An emerging technology company (referred to here as DevCo) and a large established company (LargeCo) typically have different objectives for entering partnering arrangements. DevCos generally seek these arrangements to finance growth and obtain manufacturing and distribution capabilities. LargeCos typically seek them to obtain access to new technologies and products.

Successful partnering relationships can provide strategic, operating, and financial benefits to each party. These relationships allow both parties to reach their goals more cheaply and rapidly than they could alone. In contrast, ill-conceived or badly implemented partnerships can deprive a DevCo of critical technology or markets and limit DevCo's ability to obtain future partnering or financing arrangements.

Partnering in the 1990s

While partnering relationships have existed for a long time, they are becoming even more attractive in the 1990s. DevCos are finding that the costs of technology development and product introduction have increased, while capital is more difficult and costly to obtain. Even where capital is available, the founders' equity ownership may be severely diluted. Moreover, capital alone cannot provide a DevCo with the advantages of an established manufacturing or

distribution infrastructure or the management expertise of an established LargeCo.

The current economic climate also pushes LargeCos toward partnering relationships. Many LargeCos are starved for innovation and lack the flexibility necessary to respond rapidly to technological and market changes. The disappointing results of many 1980s acquisitions have made LargeCos more leery of acquiring companies with unproven synergy. With less risk than an acquisition, partnering allows LargeCos to leverage the strategic resources needed by DevCos, such as long-term R&D budgets, manufacturing capability, distribution, and customer service.

Partnering vs. Venture Capital

When should DevCo pursue venture capital and when should it pursue a corporate partnering arrangement? When evaluating this issue, consider the following:

• *Does DevCo have proven technology?* Partnering arrangements presuppose a certain degree of technological capability inherent in DevCo. A DevCo that has not yet established its technological capability is not a good candidate for a partnering arrangement, but may well be a candidate for venture capital.

• *Is DevCo's technology divisible by market, application, or distribution channel?* If DevCo's technology cannot be divided into distinct applications, markets or distribution channels, it may be impossible to structure a partnering arrangement that gives LargeCo enough rights without significantly reducing DevCo's value. Venture capital can provide needed capital without loss of rights to technology or markets.

• *Is DevCo's cash need immediate?* Partnering arrangements tend to be slower to develop than venture capital investments. It is generally preferable to use a partnering arrangement to assist expansion rather than to ensure survival. DevCo may not have the necessary staying power and negotiating leverage in discussions with LargeCo if it needs a short-term capital infusion to meet its payroll. Such capital may be better sought from venture capitalists.

• *Do LargeCo's strategic interests conflict with DevCo's?* Where LargeCo's strategic goals conflict with DevCo's, LargeCo may have a greater conflict of interest with DevCo than a venture investor. Since venture investors seek to obtain the entire return from their investment, they are motivated to ensure the greatest possible success for DevCo. Since corporate partners obtain greater returns from commercial rights, their motivation will be to maximize the value of those commercial rights, even if that reduces the long-term value of DevCo.

• *Does DevCo need operating assistance and expertise?* Corporate partners can provide very useful operating expertise (for example, customer-driven product development, cross-licensed technology, manufacturing and distribution capability) not always available from venture capitalists.

• *Is DevCo likely to need future venture financing?* A partnering arrangement in which LargeCo acquires DevCo equity at a high valuation is attractive but may make it difficult for DevCo to raise venture capital later. Since the venture capitalist does not obtain the "operating" benefits of a LargeCo, the venture capitalist is rarely willing to pay as much for equity. Having received a higher valuation from LargeCo, DevCo may not want to accept the venture capitalist's lower valuation.

Most DevCos will combine venture capital with a corporate partnering strategy, taking advantage of the strengths offered by each at the appropriate time. The remainder of this article will help you formulate a successful partnering strategy.

Step One: Identify Strategic Objectives for Partnering

The first step in considering partnering is to identify the strategic reasons for seeking such an arrangement. What type of assistance do you need from your partner? Too often, the immediate response from emerging companies may be "money," when money is really a surrogate for services and capabilities that DevCo has not yet developed, but that may be available from LargeCo. Even when funds are needed for technology development, should it be an equity investment (with the resulting equity dilution), development funding (with the resulting technology dilution), or some combination of both?

It is equally important to identify what strategic objectives LargeCo could have for partnering with DevCo. LargeCo may seek to reduce costs, increase revenues, or improve its scheduling, quality or competitive position. A successful partnering arrangement results when the value each one obtains exceeds the costs (current and strategic) incurred in meeting its partner's objectives.

There are many different ways to look at how partnering arrangements can provide these benefits. The table "Typical Strengths of DevCos and LargeCos" focuses on typical strengths held by DevCo and LargeCo in meeting different functional responsibilities during a product's life cycle.

Understanding the Building Blocks of Partnering. To achieve DevCo's and LargeCo's partnering objectives, DevCo and LargeCo must mix and match five partnering building blocks to create an agreement that best meets those objectives. These basic partnering building blocks are:

• *Equity.* LargeCo purchases an equity stake (usually a minority position) in DevCo in return for either a seat on DevCo's Board of Directors or contractual rights to DevCo's technology. Equity purchases may be supplemented or replaced by options to purchase DevCo stock, loans to DevCo, or an option to acquire DevCo.

• *Development.* LargeCo funds DevCo's development of specific technology or products in return for rights to the resulting technology or products.

• *Manufacturing.* LargeCo receives the right to manufacture products using DevCo's technology.

• *Distribution.* LargeCo receives the right (frequently exclusive) to distribute DevCo's products within a particular geographic territory or vertical market.

• *Licensing.* LargeCo receives the right to use and modify DevCo's technology within a particular geographic territory or vertical market in a broader way than a manufacturing or distribution grant (for example, OEM agreements or internal use licenses). Licenses give LargeCo more information about the technology and typically bear royalties.

There is no typical partnering arrangement, because each is a hybrid of some or all these building blocks and can be formed during or after the development

of the technology. For example, LargeCo and DevCo may remain independent, but undertake any or all of the development, manufacturing, or marketing of the technology and products in a joint venture (by creating a new corporation or partnership). In such a joint venture, typically DevCo contributes technology rights and LargeCo contributes cash or other services. Since strategic objectives of individual companies differ, DevCo may find that seemingly similar LargeCos may be interested in partnering arrangements with very different structures.

To determine which of these building blocks is appropriate, it is important to understand what advantages DevCo and LargeCo might see in any particular building block. The table "Partnering Building Blocks" sets up a matrix of possible strategic objectives that DevCo and LargeCo may try to achieve through each building block.

Determining Strategic Needs/Bargaining Chips. Once DevCo's objectives and the likely objectives of a prospective partner have been analyzed, the next step is to list DevCo's strategic needs and likely bargaining chips. "Strategic needs" are those items of strategic value that DevCo wants to obtain. "Bargaining chips" are those items of strategic value that DevCo can offer to a potential partner. Initially, these lists should be as comprehensive as possible. Once the process of selecting specific partnering candidates begins, the lists should be pared down to those items applicable to a given potential partner. To illustrate, DevCo's strategic needs and bargaining chips lists might look like the following:

DevCo's Strategic Needs:

- approximately $5 million in capital, revenues, or reduced expenses
- distribution capability, particularly overseas or in a defined vertical market
- customer leads, credibility
- offshore manufacturing capability
- related product lines, particularly those that create an opportunity for DevCo system sales
- additional technology to incorporate into current or planned products
- sourcing of certain parts or raw materials.

DevCo's Bargaining Chips:

- existing proprietary products, technology, or patents
- R&D capability to create a unique product or component meeting LargeCo's specifications
- "influence" on DevCo's future direction, reflected by a LargeCo seat on DevCo's board of directors or R&D committee
- product manufacturing capability in the United States
- vertical distribution capability in defined areas or markets
- intangibles, such as the ability to move quickly, market focus, customer awareness, image and reputation in market niche, productivity, and trend setting ability.

Step Two: Develop the Partnering Plan

Having determined the strategic needs and the current capabilities of DevCo, the next step is to create a partnering plan. Although investment bankers or consultants may help, the key managers of DevCo must be actively involved in creating an effective plan. This plan will help

Typical Strengths of DevCos and LargeCos

Functional Responsibility	DevCo Strengths	LargeCo Strengths
Product Development	Creativity; focused development; speed; less developmental bias; flexibility	Broad market expertise; broad technical capabilities; better lab and test facilities
Legal/Regulatory	Cutting edge technology patents	Cross-license opportunities; expertise in regulatory or patent approvals
Manufacturing	Faster, low volume specialized capability; prototyping capability; specialized niche manufacturing	Component purchasing power; tooling & manufacturing; quality assurance capability
Marketing	Understanding of niche market; new technology or new market excitement; creativity; competitive edge	Broad market understanding; established product lines; volume edge; trade name recognition
Distribution	Specialized niche distribution; flexibility	Established customer base, distribution, and shelf space; better storage and transportation
Service/Support	Expertise in new technology problems	Established quality standards; warranty, service, and customer support procedures
Product Revisions	Next generation capability	Better product life extension; established user groups

maintain the focus of the advisors and management team as DevCo goes through the partnering process. A typical plan covers the following types of issues:

- strategic, operating, and financial objectives that support DevCo's strategy for increasing market share
- a strategy for developing the partnering process
- proposed deal structures, including likely alternatives, implications, and negotiating limits
- an anticipated, realistic timetable
- materials to be furnished to the potential partner. (The first information pack is to spark interest in DevCo and a second set is to provide adequate disclosure for LargeCo's initial decision regarding degree of interest.)
- a target list of partnering candidates by types, with brief reasoning and priorities.

Planning for Multiple Partnering Arrangements. Partnering arrangements are inherently more complex than a "straight" venture financing, acquisition, R&D contract, or manufacturing, distribution, or licensing arrangement taken alone. In a successful partnering arrangement, DevCo must meet the strategic objectives of LargeCo and provide LargeCo with substantial rights to DevCo's technology and market. At the same time, DevCo must ensure that it does not give away a disproportionate part of its technology and market to LargeCo in exchange for a minority equity investment or R&D contract.

When making concessions to LargeCo, DevCo must consider that it will want to do other partnering arrangements in the future. The existence of multiple arrangements adds another layer of complexity. It is often helpful to negotiate with two prospective

Partnering Building Blocks

Building Block	DevCo Objectives	LargeCo Objectives
Equity Investment	Capital infusion	Window on DevCo's technology or market
	Enhance DevCo's market credibility	Influence on DevCo's strategic direction
	Strengthen other relationships with LargeCo	Strengthen other relationships with DevCo
	Access to LargeCo's operating and market expertise	Block competitors from obtaining access to DevCo's technology
	Precursor to LargeCo's acquisition of DevCo	Precursor to acquisition of DevCo
Technology Development Agreement	Access to LargeCo's technology and patents	Access to DevCo's technology and patents
	Funding for DevCo's own R&D program	Basis for obtaining manufacturing, distribution or licensing rights
	Access to LargeCo's test and regulatory approval capabilities	Access to DevCo's faster, cheaper, or more flexible development capabilities
Manufacturing Agreement	Avoid capital expense of building own plant	Use excess manufacturing capacity
	Ship product more rapidly	Broaden manufacturing capability into new area
	Obtain LargeCo's quality and test capability	Ensure quality of product to be acquired
Distribution Agreement	Arrive at specialized vertical or geographic market faster	Complement and broaden current product line; diversify into new markets
	Get benefit of LargeCo's market reputation and support capacity	Control sales and service process
Licensing Agreement	Royalties from markets that DevCo would not otherwise address	Incorporate DevCo's technology into LargeCo's products
	Gain access to LargeCo's technology and patents	Right to change DevCo's technology itself
	Gain manufacturing and distribution capability	Right to use DevCo's technology in other parts of its operations

LargeCos simultaneously, to have an alternative when negotiations get difficult. When partnering arrangements occur in sequence, DevCo should use previous arrangements as a baseline for later negotiations. A long-term, exclusive grant of rights in one partnering arrangement may limit DevCo's flexibility in creating subsequent relationships, unless each agreement is carefully structured to fit into DevCo's strategic plan.

Step Three: Select LargeCo Targets

One of the most critical issues in the partnering process is determining which companies to approach. Enormous amounts of time and resources can be wasted with inappropriate candidates that are not likely to meet DevCo's strategic needs. Ideally, each potential candidate should meet the following criteria:

- ability to meet DevCo's strategic objectives
- strategic objectives that DevCo can meet without having LargeCo dominate DevCo's business
- clear need for DevCo's bargaining chips at the operating (not just the corporate) level
- distinct and non-conflicting target markets
- familiarity with DevCo's technologies, products, and markets
- interest, or successful experience, in making similar deals
- no recent major change in identity or organization
- compatible people and corporate cultures
- ability to complete the transaction within the required time frame.

It is difficult to obtain this type of information about potential partners, and no one partner will meet all these criteria. Still, gathering this type of information and using it to prioritize the list of candidates will both reduce the amount of time spent with inappropriate candidates and increase the likelihood of success.

Designing the Responsibility List. Once the list of prospective candidates has been prioritized, the investment banker or specific members of the management team or Board of Directors should be assigned the responsibility of contacting one or more of the top-priority targets. Ideally, the assigned person should make a personal contact within the target LargeCo who is already familiar with DevCo. Contact only a few companies at a time, both to avoid overwhelming your managers and to avoid giving the impression that DevCo is being "shopped around." As the process continues, the responsibility list should be updated to show the current contact status of each active candidate.

Foreign Versus Domestic Partners. To be a global competitor, U.S. DevCos may need to enter into partnering arrangements with foreign LargeCos. Pacific Rim or European partnerships can provide DevCo with immediate credibility, influence, and access to customers in markets that DevCo would take years of effort and losses to penetrate on its own. DevCo also may obtain manufacturing know-how, parts sourcing, and complementary technology not available to it in the United States.

Pacific Rim LargeCos may be more open to acquiring new technology and products from U.S. companies, and may be less subject to the "not invented here" syndrome than their U.S. counterparts. Moreover, many Pacific Rim LargeCos appear to have a better long-term perspective on investing in new technology and markets. Both Pacific Rim and European LargeCos view partnering with U.S. DevCos as a good way to diversify and familiarize themselves with the U.S. market and distribution channels. Such companies also tend to have experience with partnering arrangements. All of these characteristics may make foreign LargeCos more receptive to partnering proposals than U.S. LargeCos.

On the other hand, geographic distances, language barriers, and cultural differences will increase both the time and management effort needed to negotiate and implement a partnering arrangement with a foreign LargeCo. In addition to understanding and satisfying the strategic needs and objectives of a foreign LargeCo, DevCo must be sensitive to the cultural needs and expectations of its potential partner. Dealing with foreign partners also adds another layer of legal complexity, potentially subjecting the arrangement to foreign laws and regulatory approvals. From a competitive perspective, a Pacific Rim LargeCo, in particular, may prove to be a more significant long-term competitive threat to DevCo, even in the U.S. market.

Step Four: Contact LargeCo Targets

Corporate vs. Operating Staff. One difficulty in introducing a partnering proposal to LargeCo is that although LargeCo's operating staff frequently will have the greatest interest in the potential arrangement, it rarely will have the authority to engage in a significant transaction without corporate approval. For example, LargeCo's R&D department might well have an R&D budget that would allow it to pay $500,000 in advances against royalties on an R&D contract. However, a proposal consisting of a $2 million paid-up license and $3 million of equity investment would likely exceed its authority and require corporate involvement and approval. Since most LargeCos require corporate approval for an equity proposal, DevCo may want to first consider a nonequity partnering proposal.

When the operating group in LargeCo cannot complete the partnering arrangement alone, carefully consider how to approach LargeCo. If LargeCo has a corporate business development or strategic development group, DevCo should approach that group. If not, DevCo will probably want to start with or get an introduction to an appropriate person in the operating unit. Such a person can act as the champion within the corporate structure, attesting to the feasibility and value of the technology and the available market. It is critical that DevCo identify a credible and persuasive champion within LargeCo who has a vested interest in the success of the relationship. Without such a champion, partnering negotiations may eventually stall. DevCo must be prepared for the frustration, complexity, delay and re-education that results from having to work with both the operating and corporate staffs.

DevCo also should realize that the LargeCo corporate staff members who negotiate the partnering arrangement generally will not be responsible for implementing it. Successfully completing negotiations is only the first step in creating an effective partnering arrangement. Once it is clear that a partnering arrangement will be concluded, both parties must focus on how the arrangement will be implemented. While a discussion of how to effectively integrate the partnership is beyond the scope of this article, continued communications, commitment, and execution by both parties is an absolute priority to achieve sustainable success.

Confidentiality Agreements. Before attending any meetings with LargeCo, DevCo should decide exactly how much

technological, business plan, or market information it is willing to provide to LargeCo without getting a signed confidentiality agreement. The first information given to LargeCo should be sufficiently general that no confidentiality agreement is necessary. Large companies rarely sign confidentiality agreements before they have concluded that they have some threshold degree of interest in pursuing the opportunity.

Once that level of interest is attained, it is important that DevCo get a signed confidentiality agreement from LargeCo. Corporate partnering relationships invariably require DevCo to relinquish proprietary information. This is information that DevCo will not want LargeCo to use outside the scope of the proposed partnering arrangement. DevCo needs to consult with its lawyer regarding the appropriate form of confidentiality agreement and other mechanisms that can help protect its proprietary information during the partnering process. From LargeCo's perspective, DevCos frequently try to protect information that does not rise to the level requiring protection. Experienced counsel can help balance the needs of both parties and avoid having initial meetings revolve around the scope of the confidentiality agreement. In addition to confidentiality, DevCo may wish to have LargeCo agree not to solicit DevCo's key employees for some extended period of time.

Initial Meetings. DevCo can expect at least two or three initial meetings with the LargeCo staff over a period of several months, to discuss DevCo's objectives, strategy, technology, and markets. Advisors will help DevCo develop a comprehensive presentation to be given by DevCo's senior management. In these initial meetings, DevCo should be prepared to briefly outline what it wants to achieve in the proposed transaction, even though LargeCo generally will not be ready to negotiate or respond to DevCo's proposal at this stage. It is important, however, to inquire about LargeCo's strategic objectives to determine how DevCo can modify its proposal to fit the mutual objectives of the two companies.

Step Five: Create a Proposal

DevCo should expect to formulate the partnering proposal. LargeCo typically will not feel the same degree of urgency as DevCo to generate a timely and creative proposal and DevCo will want to drive the process. LargeCo's staff will help refine the proposal, establish LargeCo's negotiating limits, and assist in gaining corporate approval.

Valuing the Proposal. In formulating the proposal, consider LargeCo's strategic objectives and try to separate perceived "investment objectives" from "operating objectives." For example, LargeCo generally will purchase equity in DevCo because it is considering acquiring the company, because DevCo requires an equity investment, or because such an investment may give LargeCo some influence over DevCo, not because LargeCo wants an equity return.

LargeCo generates its returns from its operating business. As a result, LargeCo will tend to value the DevCo partnering proposal based on an operating discounted cash-flow analysis of the commercial rights that LargeCo will obtain. To determine LargeCo's cash-flow benefits from these arrangements, it is important to understand the nature of the LargeCo decision. Is this a strategic direction LargeCo cannot achieve without DevCo, or a "make or buy" decision (that is, can LargeCo accomplish its objectives faster or cheaper through DevCo than it can by doing it itself?)? A careful understanding of LargeCo's alternatives will help DevCo interpret LargeCo's discounted cash-flow analysis. Thus, DevCo should analyze the partnering proposal primarily in terms of its operating present value returns to LargeCo.

Initial Proposals. Initially, proposals should be kept in outline form, leaving the details to be developed in negotiations. Initial proposals must be realistic and businesslike. Unrealistic expectations or poorly thought-out proposals may jeopardize an otherwise promising relationship. The initial proposal should accomplish the following:

- indicate generally what DevCo wants from the transaction, leaving any equity valuation issues until later
- define what DevCo is prepared to give in the transaction, including deliverables or rights to be granted to LargeCo
- address and satisfy LargeCo's objectives and concerns
- prepare for future negotiating trade-offs
- set a timetable for completion.

If LargeCo indicates an interest in negotiating an arrangement based on the summary proposal, DevCo should prepare a detailed version of the proposal, modified to reflect what has been learned to date.

Legal, Tax, and Accounting Assistance

DevCo should seek competent legal, tax, and accounting input early in the negotiating process. Not only may laws or regulatory restrictions make some agreements impossible, but tax implications may make certain legal structures far more costly than others. In many cases, experienced counsel will know of simpler, safer, and more cost-effective mechanisms for accomplishing DevCo's objectives.

Beyond the process mentioned above, you'll also need to formulate the specific terms of your proposal, defining such things as your equity investment; terms of technology development; manufacturing, distribution, and licensing rights; and other pertinent considerations. You'll also want to address any potential international issues.

Creative Proposals for the 1990s

What type of partnering arrangements are most likely in the 1990s? A DevCo shouldn't expect early-stage equity investments or overpriced acquisitions by a LargeCo. It is far more likely that LargeCo will invest in areas that quickly generate operating returns. As a result, partnering arrangements are more likely to take the form of engineering, manufacturing, distribution, or licensing arrangements, with associated royalties and other commercial payments. DevCo must understand and plan for dilution of its rights to technology, in addition to any equity dilution.

The leverage in partnering is created by the financial and strategic resources of LargeCo and the fact that LargeCo can receive economic returns from operating benefits based on DevCo's innovations. Unlike venture capitalists, LargeCo does not require liquidity events for all its returns. Partnering is also the logical precursor to a possible LargeCo acquisition of DevCo, since it tests joint strategies and economics and provides proof in the form of results. Favorable joint economics in a partnering arrangement will increase the likelihood of a closer relationship between the two companies.

From DevCo's perspective, understanding its own objectives and the strategies of industry participants and developing an

effective partnering plan are prerequisites to successful partnering. While this process can be complicated, it is important for two reasons. First, it permits DevCo to efficiently target realistic potential partners. Second, with the assistance of experienced advisors, it permits DevCo to construct proposals composed of the building blocks discussed in this article—proposals that meet strategic objectives while leaving DevCo the flexibility it needs for the future. ♣

Jacqueline Daunt is a partner with the law firm of Fenwick & West, located in Palo Alto, California, who represents high-technology clients in domestic and international merger, acquisition, and partnering transactions. George Von Gehr is a partner with Von Gehr & Tan, an investment banking firm located in Palo Alto, California, that provides merger, acquisition, and corporate partnering services for emerging companies.

Reprinted with permission of Fenwick & West and Von Gehr & Tan.

Now Available From Apple

The following list shows APDA products that have become available to developers within the last several weeks. All prices are for the United States only; if you're outside the United States, you can obtain prices for your location by checking with your local Apple office. To get a full listing of all APDA products, check the current *APDA Tools Catalog.* For new product announcements and the most up-to-date price lists, check AppleLink (path—Developer Support:Developer Services:Apple Information Resources:APDA—Tools for Developers).

Apple Products

Apple Media Tool
B1440LL/A
$1495
Special offer until September 30—$995

Apple Media Tool Programming Environment
R0535LL/A
$2995

Apple Media Kit
(includes both Apple Media Tool and Apple Media Tool Programming Environment)
B1453LL/A
$3495

Newton Toolkit (beta version)
R0542LL/A
$795 (Note: Customers who purchase this product will be shipped the final version for free when it becomes available.)

Virtual User version 2.0.1
R0454LL/B
$150

Sound Manager Developer's Kit version 3.0
R0507LL/A
$50

Apple Internet Router Administrator's Toolkit
R0490LL/A
$199

AppleTalk Wide Area Developer's Toolkit
R0462LL/A
$50

VITAL Technical Architecture Guides
M1817Z/A
$1500

Client/Server Architecture Guide
L0211LL/A
$14.95

AppleTalk Update-based Routing Protocol (AURP) Technical Specifications
R0463LL/A
$25

LaserWriter Pro 600 and 630 Printers Developer Note
R0526LL/A
$20

LaserWriter Select 310 Printer Developer Note
R0527LL/A
$20

Macintosh Developer Note #4: Macintosh LC520, Macintosh PowerBook 145B, Macintosh PowerBook 180c
R0528LL/A
$25

Macintosh Developer Note #5: Macintosh Quadra 840AV and Centris 660AV Computers
R0529LL/A
$60

Third-Party Products

Object Master version 2.0
T0601LL/A
$395

DataPrism
T0602LL/A
$399

Inside Macintosh: QuickTime Components
T0593LL/A
$34.95

Inside Macintosh: Interapplication Communication
T0594LL/A
$36.95

Programming QuickDraw
T0603LL/A
$26.95

Symantec C++ Programming for the Macintosh
T0606LL/A
$39.95

APDA Ordering Information

To place an APDA order from within the United States, contact APDA at (800) 282-2732; in Canada, call (800) 637-0029. For those who need to call the United States APDA office from abroad, the number is (716) 871-6555. You can also reach us by AppleLink; the address is APDA. If you're outside the United States, you may prefer to work with your local APDA contact. For a list of non-U.S. APDA contacts, see the "International APDA Programs" page in the *APDA Tools Catalog.*

GetNextEvent

The indicates the trade shows/events at which Apple Computer, Inc., is scheduled to exhibit as of press time. This list may be incomplete. If you have information about a show that you want listed here, contact Apple Directions, 20525 Mariani Avenue, M/S 3034DP, Cupertino, CA 95014. For further information, check the Events folder on AppleLink (path—Third Parties:3rd Party Connection: Contests and Events).

October 17 through 20

EDUCOM
Cincinnati, OH
Contact: Eliza Lapé
AppleLink: ELIZA
(408) 974-1248

October 18 through 22

FY '94 Global Sales Conference
Las Vegas, NV
Private event by invitation
Contacts:
Marva Whelan (408) 974-8561
Gail Bridges (408) 974-3094
Al Hoodwin (408) 974-4419
Thad Carhart 33-1-4901-4720
AppleLink: WHELAN2, BRIDGES2 (Apple USA), HOODWIN (Apple Pacific), CARHART (Apple Europe)

October 27 through 29

ITTE/NSBA
Dallas, TX
Contact: Jeryl Gerhardt
AppleLink: JERYL
(408) 974-2368

October 28 through 30

Seybold
San Francisco, CA
Contact: Russ Havard
AppleLink: HAVARD1
(408) 974-4371

November 7 through 10

MacIS Conference "Back to the Future"
Orlando, FL
Contact: Jerry Star
AppleLink: JERRY.STARR
(408) 974-3836

November 12 through 16

National Alliance of Realtors (NAR)
Miami Beach, FL
Contact: Cheryl Bunch
AppleLink: BUNCH1
(408) 974-2853

November 14 through 17

League for Innovation
Nashville, TN/Apple in attendance
Contact: Jeryl Gerhardt
Apple Link: JERYL
(408) 974-2368

November 15 through 19

Comdex
Las Vegas, NV
Contact: Dave Billmaier
AppleLink: BILLMAIER1
(408) 974-6553

December 6 through 7

International QuickTime & Multimedia Conference
Paris, France
Contact: Sandy Butler
AppleLink: BUTLER.S
(415) 904-0811

December 7 through 10

CAUSE93
San Diego, CA
Contact: Eliza Lapé
AppleLink: ELIZA
(408) 974-1248

December 9 through 10

International Quick-Time & Multimedia Conference
Munich, Germany
Contact: Sandy Butler
Apple Link: BUTLER.S
(415) 904-0811