



AppleDirections

Inside This Issue

Editor's Note: Hardware Wars	2
IndustryWatch: Advantage Apple?	3
Getting Started With Netware 4.1 NLM Development	8
System 7.5 to Debut this Fall	9
More Than 130 Native Power Macintosh Applications Shipping	10
New Developer Note	12
CD Highlights: Tool Chest Edition	12
PCI Cards for the Second Generation of Power Macintosh	13
Developer University Schedule	15
Human Interface	16
Marketing Research Monthly: Macintosh Attracts New Users	20
Marketing Feature: Product Reviews	21

Apple Developer Central at Boston Macworld
 If you're attending the Macworld Expo show at Boston's World Trade Center August 2-5, be sure to drop by Apple Developer Central. Located in the Apple Pavilion, Apple Developer Central is a focal point for developer-related products and services. There will be a tools demonstration area where you can learn more about new programming products from Apple and third parties; a theater where you can enjoy presentations and tutorials on a variety of technical topics; and an APDA area where you can take advantage of special show prices. ♣

Apple News

eWorld, the New On-Line Community

Last month, Apple introduced its new eWorld on-line service in a different way—on eWorld. Reporters all over the world logged onto eWorld and participated in an electronic “event,” complete with presentations from Apple officials (including Apple CEO Michael Spindler) and a question-and-answer session with the press.

When users connect to eWorld, they first see a rendering of a town with different buildings that house different kinds of information and services. Users receive their mail by clicking the eMail Center, which is an electronic post office; for historical research materials, they'd go to the Learning Center. Other buildings are the Marketplace, Newsstand, Arts & Leisure Pavilion, and the Computer Center. The eWorld service also allows subscribers to exchange text messages with Internet users as well as with subscribers to other services accessible through the Internet.

Currently, the service includes eWorld for Macintosh and NewtonMail, a messaging service for Newton. Like an “on-line town square,” eWorld for Macintosh provides a global electronic mail system together with news, information, and other services from about 100 partners such as Reuters America, Inc., Tribune Media Services, ZiffNet/Mac, and Inc. Magazine Online. Apple intends to release

please turn to page 8

Strategy Mosaic

The Future of Development Tools

Apple's New Strategy
By Gregg Williams, Apple Directions staff

In the beginning was MPW, and it was good enough—or so Apple said. (Yes, I know—the Lisa Workshop came first, but humor me.) Apart from Apple's own Macintosh Programmers Workshop (MPW), a few third-party compilers struggled for a while. But the vendors of these compilers didn't have access to all the information Apple Computer, Inc., had, and not enough developers existed to keep them in business, let alone pay for the development they needed to improve their products. So if Apple was going to ensure the success of the Macintosh platform, it had to deliver development tools, or developers wouldn't be able to beget any software.

Slowly, things changed. As the Macintosh platform became more successful, the number of developers increased enough to support one development tools company, then several. Developers wanted more languages, more features, more support—and Apple couldn't provide it all. Finally, Apple has recognized the same trend with development tools that it's seen for a long time with application software: Apple can't do it all alone, so it needs to ally itself with outside companies that can help.

please turn to page 4

AppleDirections

Volume 2, Number 8

Apple Directions, the monthly developer newsletter of Apple Computer, Inc., 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 Apple Developer Periodicals group within Apple's Developer Press.

Editor

Paul Dreyfus (AppleLink: DREYFUS.P)

Technical Editor

Gregg Williams (GREGGW)

Business & Marketing Editor

Dee Kiamy (KIAMY)

Associate Editor

Anne Szabla

Production Editor

Lisa Ferdinandsen (LISAFERD)

Contributors

Pete Bickford, Alex Doshier, Sally Harris, Timothy Hughan, Sandee Karr, Kris Newby, Robert Patrick, George Towner, Jessa Vartanian, Allen Watson

Manager, Developer Press

Dennis Matthews

Manager, Developer Support Information

Greg Joswiak

Production Manager

Diane Wilcox

PrePress/Film

Aptos Post

Printer

Wolfer Printing Co., Inc., Los Angeles, CA

© 1994 Apple Computer, Inc., 1 Infinite Loop, Cupertino, CA 95014, 408-996-1010. All rights reserved.

Apple, the Apple logo, APDA, AppleLink, AppleTalk, HyperCard, Lisa, MacApp, Macintosh, Macintosh Quadra, MacTCP, MPW, Newton, and PowerBook are trademarks of Apple Computer, Inc., registered in the U.S. and other countries. AOCe, AppleScript, AppleSearch, AppleSoft, DocViewer, Dylan, eWorld, Finder, MessagePad, NewtonMail, OpenDoc, Performa, PhotoFlash, PlainTalk, PowerBook Duo, Power Macintosh, PowerTalk, QuickDraw, QuickTime, True Type and ResEdit are trademarks of Apple Computer, Inc. Adobe Dimensions, Illustrator, Photoshop, Postscript, Premiere, and Adobe Type Manager are trademarks of Adobe Systems Incorporated, which may be registered in certain jurisdictions. NuBus is a trademark of Texas Instruments. PowerPC is a trademark of International Business Machines Corporation, used under license thereof. Windows is a trademark of Microsoft Corporation and SoftWindows is a trademark used under license. UNIX is a registered trademark of UNIX Systems Laboratories, Inc. All other trademarks are the property of their respective owners.

Mention of products in this publication is for informational purposes only and constitutes neither an endorsement nor a recommendation. All product specifications and descriptions were supplied by the respective vendor or supplier. Apple assumes no responsibility with regard to the selection, performance, or use of the products listed in this publication. All understandings, agreements, or warranties take place directly between the vendors and prospective users. Limitation of liability: Apple makes no warranties with respect to the contents of products listed in this publication or of the completeness or accuracy of this publication. Apple specifically disclaims all warranties, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

Editor's Note

Hardware Wars

It's not unheard of in the high-tech business for companies—especially market leaders—to announce new technologies simply to keep customers from buying the competition's product. The reasoning goes something like this:

"Gee, XYZ Tech, Inc., just shipped its computer with the brand-new FST architecture and people really like it. Gosh, we're not even close to having FST in our systems yet. Sooooo—let's announce we're working on FST. No, make that we're working on something better than FST.

"Great! Then customers won't buy the XYZ Tech FST systems, thinking they'll wait for ours, instead. And, when XYZ Tech's FST strategy fails, maybe we don't even have to deliver our FST solution.

"When can I call a press conference?"

That strategy is one of the things that came to mind when I read the recent joint Hewlett-Packard/Intel announcement about the RISC (they're even calling it "post-RISC") personal computer microprocessors the two companies will apparently collaborate on.

It also occurred to me that Hewlett-Packard and market leader Intel are giving Apple Computer, Inc., and its Power Macintosh computers a real vote of confidence. Seems like just yesterday that Intel was trying to woo Macintosh users to Intel Pentium-based PCs (which rely on CISC architecture) with ads suggesting that Apple was headed in the wrong direction with its RISC PowerPC processor-based Macintosh computers. With their announcement, Intel and Hewlett-Packard are apparently agreeing that RISC is the way of the future.

Too bad it will take them so long to actually ship their silicon; the announcement implies that it could be the end of the century before the Hewlett-Packard/Intel collaboration bears fruit. By then, Power Macintosh computers—today's only RISC-based personal computers that use a mainstream operating system—will have been on the market for years.

The Hewlett-Packard/Intel announcement, even with its "promise" of delivering post-RISC technologies, such as very long instruction flow (VLIW), doesn't seem like

cause for worry. It will be years before the first products using the technology are available. In the meantime, Apple expects to make its Power Macintosh computers more powerful and the Macintosh operating system even easier to use than it is today. By the end of the century, Apple expects its RISC computing strategy to have paid off in significant gains in market share.

What can you do? Just what we've been telling you to do for some time: If you haven't already, take your product "native," porting it to run in native PowerPC mode so it can join the 130-plus native Power Macintosh applications currently shipping (see the news story on page 10).

Next-Generation Power Macintosh Computers

As if having computers that provide industry-best price/performance weren't enough: No sooner did Apple release the Power Macintosh computer than it started working on its next-generation RISC-based systems, which will provide even greater performance than the current computers, and not just because of faster PowerPC chips.

The new computers, due sometime next year, will employ the PCI (Peripheral Component Interface) bus and Open Firmware standards, which are expected to provide enhanced performance, greater user flexibility because they'll be able to run multiple operating systems, and a wealth of developer opportunities. More than 400 developers at the 1994 Worldwide Developers Conference said they are planning cards and drivers that support the new standards; you'll want to read George Towner's article on page 13 to decide if you want to join that number.

Just as we did with PowerPC, we're telling you about the new technology early, because we want to be sure that, when Apple raises the hardware technology bar to its next level, you're prepared to make the leap.

Paul Dreyfus
Editor

IndustryWatch: News & Perspective

Advantage Apple?

Prepared by the Apple Directions staff

IBM PowerPC Offering Likely Not to Be Early Competition for Power Macintosh Systems

IBM is expected to roll out its first PowerPC processor-based personal computers this fall, but it's not clear how well they'll be received in the marketplace, as the following reports suggest.

- *Workplace OS slow to arrive.* Trade press reports suggest that Workplace OS, which will run OS/2, Windows, and DOS applications, will not appear for at least several months after the PowerPC processor-based personal computers arrive. Reportedly, the first IBM PowerPC processor-based machines will only run AIX, IBM's version of the UNIX® operating system, and a ported version of Windows NT. When Workplace OS arrives it's expected to run today's 32-bit OS/2 applications, Windows applications, and DOS applications in emulation. Sixteen-bit OS/2 applications apparently won't run at all; it's anticipated that AIX applications will run on Workplace OS on top of an AIX "personality" that will appear at some point.

The only applications that will run at true PowerPC speeds are recompiled 32-bit OS/2 applications and AIX applications. Since there are comparatively few 32-bit OS/2 applications available today, even porting 100 percent of them to Workplace OS will not guarantee success for IBM's first PowerPC processor-based products.

- *PowerPC line to be delivered late and won't have its own identity.* Originally planned for release this summer, the first models now won't be released until late 1994, according to the *Wall Street Journal*. In the same story, the *Journal* reported that IBM PowerPC processor-based systems will not be given their own identity; instead, they'll be one of several types of PCs sold by IBM's normal PC business unit.

- *AIX not PReP-compliant.* IBM's PowerPC Reference Platform (or PReP) specification is supposed to be the company's guidebook for building PowerPC processor-based hardware and operating systems. However, PReP operating systems are supposed to have an abstraction layer between themselves and the hardware. AIX runs on the hardware itself, so it's not PReP-compliant. Workplace OS, when it arrives, is expected to be PReP-compliant.

Implications/Opinions: When the first IBM PowerPC processor-based machines ship, they'll run only AIX and Windows NT, and few native applications will be attractive to many users. Because of these factors, we think they'll be suitable for use only as UNIX workstations and servers. In fact, it will be a stretch to label a PowerPC processor-based machine running AIX or Windows NT as a "personal computer"; users will need too much memory to run the operating system (at least 16 MB) and will have access to too few productivity applications for the computers to be considered "true" PCs. We think IBM will not achieve PC-style sales with its PowerPC systems, or economies of sale, until it ships a personal computer-style operating system for the computers and a variety of native applications are available.

IBM has said that Workplace OS with an OS/2 personality is supposed to run in 8 MB of memory, and that it's supposed to ship by the

end of the year. However, Workplace OS is based on Mach, a microkernel OS, and Mach may be too resource-hungry to form the basis of a mass-market operating system. Further, since Workplace OS is a completely new operating system (that hasn't even reached beta-test phase), the end of the year may be an optimistic ship date. Also, we wonder how quickly application developers, already straining to move their applications to Power Macintosh and Chicago, will be able to divert resources to undertake Workplace OS development.

The first IBM PowerPC processor-based machines will probably attract a lot of attention, but we doubt that they'll be serious competition for Apple's Power Macintosh systems or IBM-compatible, x86-architecture machines this year. While IBM may have early success with workstation users, and in the server market, successful entry into the personal computer arena will be, at best, a long-term proposition.

Pentium Sales Slow

A number of articles in the press, including one in the *Wall Street Journal*, have reported that Intel's Pentium processors are selling more slowly than the company expected and that, to boost Pentium sales, Intel has launched a \$150 million marketing campaign. Also, according to Computer Intelligence InfoCorp., PowerPC processor-based Power Macintosh computers are outselling Pentium-based PCs in the United States by more than 50 percent.

Implications/Opinions: We think it's far too early to celebrate Pentium's failure. Some of the negative press may be based on simplistic analysis. Intel expects to sell between 5 and 7 million Pentium processors this year, but appears to have sold fewer than 1 million in the first six months. However, Intel has always planned to sell the majority of the processors late in the year, so it may not be as far off the sales mark as it currently appears.

Doesn't Feel Like "Plug and Play" to Us

You're going to be hearing a lot about Plug and Play, the "new" standard being adopted by manufacturers of IBM PC-compatible computers. The Plug and Play standard is intended to make it easier for PC users to install and use new hardware devices and software.

Lest you fear that the Macintosh computer will lose its advantage in this area, we're pretty sure that, even compared to a new Plug and Play PC, the Macintosh will still be significantly easier to use. For example, here are the steps required to add a Plug and Play CD-ROM drive to one of the currently easiest-to-use Plug and Play PC compatibles. (We won't mention the manufacturer by name to protect the innocent.)

1. Turn off the PC.
2. Disconnect the power cord from the wall outlet.
3. Disconnect any other external equipment connected to the PC.
4. Using a Torx T-15 screwdriver, remove the two screws in the back of the PC.
5. Slide the system tray out of the PC.
6. Locate an unused expansion slot.
7. Remove the screw at the top of the expansion slot, and then lift the expansion slot cover plate out of the PC.

8. Lower the adapter card into the expansion slot and press it firmly into place.
9. Replace and tighten the screw that attaches the card's bracket to the PC chassis.
10. Slide the system tray into the PC.
11. Replace and tighten the two screws in the back of the PC.
12. Plug the CD-ROM drive into the adapter card.
13. Plug the external equipment back into the PC.
14. Plug the power cord into a wall outlet.
15. Turn on the PC.

Here are the steps required to add a CD-ROM drive to a Macintosh computer today:

1. Turn off the Macintosh computer.
2. Plug the CD-ROM drive into the SCSI port.
3. Turn on the Macintosh computer.

Implications/Opinions: The current Macintosh computer is already easier to use than the competition's future standard. Future hardware and software releases from Apple are expected to make the Macintosh interface even easier to use, providing increasing levels of control and assistance to the user. Keep on doing what you're doing with Macintosh applications and we can expect the Macintosh platform to maintain its ease-of-use advantage.

Intel Pushes Pentium, Restricts 80486 DX4 Processor

Intel has quietly notified its other equipment manufacturers (OEMs) that availability of 100-MHz DX4 80486 microprocessors will be very limited. At the same time, it lowered prices for 60-MHz Pentium chips.

Many OEMs and end-users would have found the DX4 to be quite appealing. The performance levels of the two processors are very similar, and systems based on the "cooler" DX4 are both easier and less expensive to manufacturer than those based on the "hot" Pentium part. But if 100 MHz DX4s are made unavailable and 60-MHz Pentium parts are priced low enough, Intel can shift the market away from DX4s and toward the low-end Pentium.

Implications/Opinions: Even though Intel could make more short-term profits by selling DX4s rather than the 60-MHz Pentium chips, which are expensive to manufacture, Intel wants to shift the 80x86 market from 486 to Pentium as quickly as possible. AMD, Cyrix, IBM, and others will all be offering high-clock-speed 486 processors by the end of the year, so Intel needs to move the desktop market to Pentium before its margins erode further from 486 price wars. By the time the market is convinced Pentium is a better choice than 486, Intel should be ramping up production of its P54C Pentium processors, which are less expensive to manufacture.

Cairo Delayed

Microsoft conceded at NetWorld/Interop in May that Cairo, the next major release of Windows NT, won't go into beta testing until 1995. Cairo is expected to be a high-end object-oriented operating system.

Implications/Opinions: This is good news for OpenDoc, because Cairo is a version of Windows NT that's supposed to distribute OLE objects across the network. As long as Cairo is delayed, OpenDoc—which is expected to be network-ready on its initial release late this fall—should have a competitive advantage. ♣

Strategy Mosaic

The Future of Development Tools

continued from page 1

But there's another wafer-thin detail: In the past few years, Apple's margins have been sliced to the bone—from over 50 percent to under 25. The once-overflowing coffers that funded new development tools and technologies are no more. Where is AppleSoft (the division of Apple responsible for all Apple software products, including development tools) going to get the money to continue its innovation?

This Ain't Kansas, Toto

Apple is changing the way it will get development tools into the hands of anyone who wants them, and that's what this column is about. I caution you not

to write this column off as marketing effluvia. It's based on two sources: the keynote speech on development tools at last May's Apple Worldwide Developers Conference (given by Ike Nassi, vice president, AppleSoft Development Products), and the definitive cross-my-heart-and-hope-to-die-Apple-Confidential internal document that served as the basis for Ike Nassi's WWDC speech. Granted, there are some things Better Left Unsaid, but I think that what follows is an accurate overview of Apple's actual development tools strategy. If you can find room for any hidden agendas here, I have a great conspiracy theory to sell you. . . .

Apple's development tools strategy can be summarized in three points:

- first, to give you a wide choice of traditional development tools at a reasonable price
- second, to develop and

make available new development tool technologies that will give you a quantum leap in your productivity

- third, to provide cross-platform development tools that allow you to reach the Macintosh and Windows platforms

Before I describe the details of Apple's plan for development tools, let me say a bit more about the first two points, because they provide the foundation for the specific steps Apple will be taking.

Traditional Tools, Reasonably Priced

As I stated earlier, Apple has realized that it can't be all things to all developers. To try to do so would be to fail, so Apple has realized that it needs to concentrate on what it does best and leverage the results. And what is it that Apple—and only Apple—can supply? The core development

technologies that are closely linked to the Macintosh architecture, and new development technologies that only Apple has the resources to develop.

Apple's strategy for core development tools—compilers, linkers, debuggers, and such—is as follows. First, Apple will continue to provide "reference" core development tools, basic tools that can always be used to "get the job done," no matter what the job is. Second, Apple will license key development technologies to third parties, who will then produce various development tools wherever they see a market. This strategy has several positive results:

- You will get a wider choice of tools, and you can select the ones that best meet your needs.
- Commercial core-tool developers will provide the best developer support possible.
- Apple will provide reference

core tools and improve them for the benefit of current users.

- Apple will also be able to concentrate on what only it can do: innovate in and strengthen the development tools platform.

I should explain what I mean by “reasonably priced.” Apple’s core tools will be reasonably priced because Apple will provide them as inexpensively as possible given the current financial realities. *Apple doesn’t expect to recoup its development costs from selling traditional development tools.* Also, third-party tools will be reasonably priced because of market competition.

Developing—and Funding—New Technologies

Apple is about nothing if not innovation. Its products have changed the way people write, draw, communicate, run their businesses—but *you* are largely programming the same way you were ten years ago. What’s worse, with programs getting more complex, software development has become a bottomless pit for your time, and you can’t bring products to market fast enough.

Apple has decided it’s time for some “breakthrough” innovation in software development. Part of Apple’s strategy is to fund development of new tools and new releases of existing tools, allowing you to be more successful. These developments will enable you to

- create products you couldn’t create before (which will give you additional revenue)
- create products more quickly and easily (so you can do more, and make more money from additional products)
- reach other platforms with minimal extra effort (so you can sell to more customers and make more money)

But here in the land of 25-percent margins, the money has to come from somewhere new. Apple doesn’t want to strangle

basic development, but part of Apple’s plan is to make a reasonable profit from new tools that add value—that is, tools that help *you* make more money than you could otherwise. Apple also plans to make some of the money it needs by selling users innovative products based on the technologies it is putting in place today (more on this below).

But Apple doesn’t just need money—it needs the security of knowing it has a stable source of income in the long term. Such security will make it safe for Apple to plow a good part of its profits back into research and new development tool technologies, which will ensure future profits and future security. This will complete the cycle for Apple *and* provide you with development tools that will help make you successful.

OpenDoc is a key factor in ensuring Apple’s security and future success. So far, Apple has had to compete with an established standard that it does not control. With OpenDoc, Apple and its business partners within the Component Integration Laboratories (CILabs) are building a cross-platform horizontal layer that *nobody* owns. This will create a level playing field in which Apple and OpenDoc developers can more successfully compete.

Since Apple developed many of the technologies that go into OpenDoc, it will be in a unique position to supply cross-platform tools, OpenDoc parts, and end-user solutions based on OpenDoc. OpenDoc adds value, and Apple plans to profit from the value it has added.

Of course, OpenDoc won’t make any money for Apple if you don’t use it—but Apple believes that you will. OpenDoc was designed to give you compelling reasons to adopt it: It will lower your development costs, level the playing field (to allow small companies a chance to compete in the market), open up new prod-

uct categories, decrease your product support, and enlarge the customer base for your products. (For more information, see pages 4–5 of last month’s column, “The Best WWDC Ever?”) Also, users will want OpenDoc-based software because it will give users more control over their work environment, and it will make their computers easier to use.

One Size Does Not Fit All

From here, I want to go into the details of Apple’s immediate development tool plans. But first, I need to tell you how Apple has structured its thinking about development tools.

Apple expects many people to use tools—and the easier they are to use, the more people Apple expects to use them. (From this, you can infer that Apple plans to make money from tools that allow average users to create their own software solutions—there are a lot more average users out there than commercial developers, and many of them have annual software budgets.)

Apple has traditionally served commercial developers, who most commonly use development tools to create “horizontal” applications of interest to a large number of people. But other people want to use computers to solve their problems, too. In-house developers must quickly create specialized software for their company and modify it several times during its lifetime. Consultants, value-added resellers (VARs), and systems integrators (SIs) must create customized solutions for their clients within a wide range of time and money constraints. “Power users” (including businesspeople, teachers, and students) would rather create their own solutions—if tools are available that make it easy enough to do so. And a new but growing group, content authors, have information they want to access from a computer.

Apple divides development tools into three areas:

- core tools (for example, compilers, linkers, and debuggers)
- application architectures (OpenDoc) and frameworks (MacApp)
- solution tools (HyperCard, AppleScript)

The sections that follow describe what Apple has done and what it plans to do in each of these three areas.

Core Tools

Without core development tools, you can’t create new software. So Apple planners knew that the success of the recently introduced PowerPC processor-based Macintosh computers depended as much on the availability of PowerPC development tools as it did on that of the hardware itself. By the March 14 Power Macintosh introduction date, Apple was shipping the Macintosh on RISC Software Developer’s Kit (SDK); this is an MPW-based C/C++ cross-compiler that allows you to develop on 680x0-based Macintosh computers and run the compiled code on a Power Macintosh computer.

Your feedback to Apple on MPW indicates that you find it “industrial-strength” (you like its scripting, extensibility, and robustness) but that it compiles much too slowly. For those of you who use MPW as your primary development platform, Apple is working on making the next version of MPW, due in the second half of 1994, compile faster. In later versions of MPW, more MPW tools will be converted to native PowerPC code, which should also speed things up.

Other Core Tools

Since MPW may not meet everyone’s needs, Apple is working with third-party developers to bring other PowerPC develop-

ment options to market. Apple worked closely with Metrowerks, Inc., which released both 680x0 and PowerPC versions of CodeWarrior before the first Power Macintosh computers were available. CodeWarrior currently includes C and C++ compilers and, on the 680x0 side only, a beta version of an MPW 3.3 Pascal-compatible compiler (a PowerPC version is due in August). Additionally, Metrowerks plans to introduce 680x0 and PowerPC versions of Object Pascal compilers by the end of 1994. Language Systems also sells a 680x0 Object Pascal compiler and is working on a PowerPC version.

Why the interest in Object Pascal? Before version 3.01, the MacApp framework used Object Pascal rather than C++, and some developers have a lot of code written in Object Pascal. Though the Power Macintosh architecture is different from that of the 680x0, it may be possible for developers to update their existing Object Pascal MacApp programs to work on Power Macintosh computers. This is the primary reason that developers have asked for a PowerPC Object Pascal compiler.

Symantec is shipping a “developer’s version” (read: beta) of a PowerPC cross-compiler for Symantec C++ 7.0. With help from Apple, Symantec is working on a native PowerPC development system. Absoft is shipping a PowerPC FORTRAN compiler.

Slightly further ahead (with milestones throughout 1995) is Apple Dylan, a “dynamic language” that Apple thinks will give programmers the high productivity of object-oriented dynamic languages such as Smalltalk or Lisp, and the high run time performance and commercial viability of languages such as C and C++. Internal developers at Apple have reported finishing projects two to three times faster than with C++.

Dylan automatically manages memory allocation for you, a task that, by some estimates, takes up to 40 percent of a programmer’s time. And you’ll be able to use Apple Dylan with existing C and C++ code and application programming interfaces (APIs). Apple is working on a framework written in Dylan that will make application development even easier. To meet the needs of developers targeting multiple platforms, work is underway on a version of Dylan for Microsoft Windows.

Application Architectures and Frameworks

Things change, and failing to recognize—and act on—change can be a costly mistake. When Apple and Symantec started the Bedrock cross-platform development environment in mid-1992, the idea of software parts was barely on the horizon. Apple

further development from Symantec.

Apple is combining the Bedrock and OpenDoc technologies to create the OpenDoc Parts Framework (OPF), which will allow you to create commercial-quality OpenDoc parts for both Macintosh and Windows with one development effort. Apple strongly recommends that you use the OpenDoc Parts Framework for future development because it will give you the increased productivity of a framework, full entry into the world of OpenDoc software, and cross-platform development capabilities, all in one. An early development version of the OpenDoc Parts Framework was given to attendees of this year’s WWDC, and a later version will be distributed more widely with versions of OpenDoc due later this year. Apple expects the first commercial version of

Doc containers, PowerTalk mailers, and scripting—and will extend the life of today’s C++ MacApp source code. OpenDoc container support, which allows a document created by your application to be a container for other OpenDoc parts is a key MacApp 3.5 feature. MacApp 3.5 does not provide a full OpenDoc solution; it still creates a monolithic application, not a part that can be embedded in another part. However, it will allow a large number of developers to participate in (and contribute to) the acceptance of OpenDoc.

I can’t state this too clearly: *MacApp is an interim solution that will not be supported in the long term.* Apple’s enhancements to MacApp are meant to give current MacApp users some “breathing space” in which to make the transition to a fully component-based implementation of the OpenDoc model.

Apple recommends that, when it becomes available, you use the OpenDoc Parts Framework. Doing so will probably necessitate a major revision of your application, but it will greatly increase the life span of the resulting source code and give your software two important things—full OpenDoc support and a cross-platform solution.

Third-Party Frameworks

Several third-party companies also provide frameworks. (I’m including products with extensive class libraries, which do much of the work of a framework.) XVT Software, Inc., offers tools that help you define a user interface that is portable across the Macintosh, Windows, and UNIX platforms.

Today you have your choice of core tools that are bundled with their own frameworks. Symantec’s C and C++ compilers include the THINK Class Library. The Metrowerks CodeWarrior compilers include PowerPlant, a

OPF will give your software two important things—full open Doc Support and a cross-platform solution.

realized that by the time the Bedrock project was about to come to fruition, the last thing you wanted to hear was “Here’s a framework that creates monolithic applications—but that’s a total dead end, and we want you to start creating OpenDoc parts about six months from now.”

The decision to stop work on Bedrock was difficult to make, and Apple knew that doing so would justifiably anger some people in the developer community. However, staying with Bedrock (and perhaps killing it even later) would have been even more destructive to both Apple’s and your fortunes, so Apple ended the Bedrock project last January and licensed the rights to

the OpenDoc Parts Framework to ship in mid-1995.

The switch, mid-stream, from Bedrock to the OpenDoc Parts Framework would have left you without a viable application framework. That, plus developer feedback and a need to support Power Macintosh computers and OpenDoc, led Apple to re-animate MacApp. The current version of MacApp, version 3.1, is on the Macintosh on RISC SDK and the current E.T.O. (Essentials-Tools-Objects) CD, and it will allow you to compile a MacApp application for either the 680x0 or the PowerPC processor.

MacApp 3.5, due in 1995, will support several key technologies—primarily support for Open-

framework that creates scriptable, recordable applications.

A number of core tools that include frameworks are also cross-platform. Digitalk, Inc.'s Smalltalk/V is available for the Macintosh, Windows, and OS/2 platforms. Quasar Knowledge Systems sells a Macintosh version of its SmalltalkAgents language/development environment; it plans to have cross-platform versions for Power Macintosh, Windows, Windows NT, and UNIX by the end of 1994. Prograph International sells a Macintosh version of Prograph CPX; it plans cross-compilers and native development environments for Windows, MS-DOS, and PowerPC processor-based systems.

(Products from all of these companies except Quasar Knowledge Systems are available from APDA. You can reach Quasar at 301-530-4853; see the back page of this issue for APDA phone numbers and ordering information.)

One framework that deserves special mention is the result of an alliance between Apple, Borland International, Inc. (with its Object Windows Library, or OWL), and Novell, Inc. (with its AppWare Foundation). They are combining their technologies to create ObjectWindows for AppWare Foundation, a C++ cross-platform application development library for the Macintosh, Windows, and UNIX platforms.

A future version of ObjectWindows for AppWare Foundation will work in the OpenDoc world. At the WWDC, Novell demonstrated using the product to build OpenDoc parts, and the company said it would have "a full-fledged parts framework for OpenDoc later this year."

You can order the AppWare Foundation Software Developer's Kit from APDA, and Novell is currently shipping a beta version of ObjectWindows for AppWare Foundation. For information on

Novell's AppWare products (in the U.S.), call 800-277-2717; to order AppWare products, call 800-733-9673.

The bottom line on frameworks is that different ones are available, and you must choose the one that fits your development history, resources, target markets, and goals. Apple recommends that you use MacApp now and the OpenDoc Parts Framework later. If you want your solution to be portable across platforms and are willing to stay with the monolithic application format, you might choose ObjectWindows for AppWare Foundation. Depending on other constraints, you might go with some other framework.

Solution Tools

Solution tools allow consultants, integrators, VARs, and others to create computer-based solutions. Apple currently supplies several solution tools—HyperCard, AppleScript, and the Apple Media Tool.

HyperCard (which I've always believed to be as fundamental a piece of software as the original VisiCalc spreadsheet) has reached new heights with version 2.2. With its color and AppleScript support, adherence to the Macintosh human interface guidelines, and its ability to make stand-alone applications (I wish they were smaller, though), HyperCard 2.2 is a far more capable solution tool than most developers think.

Tens of thousands of businesspeople, consultants, educators, and students are using HyperCard 2.2 to finish projects much faster than they otherwise could have. It surprises a lot of people to know that many Fortune 500 companies use HyperCard for in-house solutions and that some of today's most innovative titles, like Broderbund's Myst and MacPlay's Xplora 1: Peter Gabriel's Secret World, were created using HyperCard.

At a higher granularity, AppleScript allows users to "wire" existing scriptable applications—over 100 applications are scriptable today—to create custom solutions for (usually) very specific jobs. (Not that AppleScript is just for custom solutions—you can add powerful new features to your commercial program by coding them in AppleScript.)

The AppleScript Scriptor's Kit 1.1 also includes the FaceSpan Interface Processor, which gives you an easy way to create a user interface for the scripts you've written. By making use of the scriptable Finder (included in the AppleScript Scriptor's Kit 1.1, as a standard feature of this fall's System 7.5), you can also use the Finder to carry out file-related tasks for you.

If your project is heavy on multimedia or content, the Apple Media Tool (and, for heavy-duty use, the Apple Media Tool Programming Environment) may be the solution tool you'll want to use. It's well suited to large, multi-person projects and uses a visual programming language, so non-programmers can use it. Third parties have already used the Apple Media Tool to create commercial CD-ROM products; one of its biggest advantages is that you can use it to generate both Macintosh and Windows products from the same project.

Future Solution Tools

The future—it's a place where everyday people will be their own programmers, and all of you will be out of a job. . . . Just kidding, folks; you've still got plenty of job security. But Apple *is* planning future solution tools that will offer consultants, VARs, and others a more powerful, extensible development environment with an ease of use comparable to that of today's HyperCard. In the future, Apple hopes to make some kinds of programming a lot easier.

On the subject of exactly what these solution tools will be, my crystal ball says, "Reply hazy—try again later." Still (according to Ike Nassi in his WWDC speech), you can be sure that future solution tools from Apple will

- work well with OpenDoc
- provide opportunities for Apple and third-party additions
- allow you to create Macintosh and Windows versions from the same development effort
- provide solutions that have reasonable disk and memory footprints

Getting There From Here

I'm fond of saying that most problems are not, in theory, so difficult to solve—it's solving them from where we are *right now* that's difficult. And that's the case with Apple's development tools strategy: If Apple could start from scratch, it could come up with a much better strategy that would make some people a lot happier. But it can't—Apple has to start with where it is *today*, with the lines of code you've already written and the preferences of the buying public.

Given the reality of reality, the development tools strategy I've outlined here is, I think, realistic. What's your opinion? What could Apple do to improve it? Write me at AppleLink address GREGGW or greggw@apple.com (Internet); I'll forward your input to the right people in Apple. ♣

Apple News

eWorld

continued from page 1

a version for DOS/Windows-based personal computers in 1995.

One thing that distinguishes eWorld from other on-line services is the attractiveness and simplicity of its interface. In fact, *Wall Street Journal* personal technology columnist Walter Mossberg called eWorld “the most graphically attractive and easiest to use of the current services.” Mr. Mossberg, who was one of more than 10,000 beta testers, also wrote that eWorld will “reduce the sense that this is a typical computer environment,

and . . . give users the feeling of a friendly electronic community.”

For you, eWorld is not just a place to communicate, learn, and be entertained—it’s a way for you to reach your customers. You can use eWorld to market, support, and distribute your products using one of two methods. First, for a one-time charge of \$500, you can purchase a Vendor Booth, an interactive forum that allows you to exchange information with your customers. You can use your Vendor Booth to post product news, demos, press releases, new versions of your products, and any other on-line material that would be useful for your customers. Additionally, you can use it to resolve support issues and

provide discussions between you and your customers.

Your second option for using eWorld is the Showcase, a one-way posting area where you can publish information, including data sheets, promotional information, and demo files. Posting on the Showcase is free.

In addition to Apple, many key industry publications and services as well as leading software and hardware vendors have posted their latest support and product information in eWorld. These companies include Claris Corporation, Supermac Technology, CE Software, and Global Village.

According to Arwyn Bryant, who manages developer participation on eWorld, this is just the

start. “We want eWorld to go beyond conventional expectations for developer activity,” says Bryant. “We are developing Open API [application programming interface] and client toolkits that will let developers build modules that connect users to on-line information, upgrades, or product add-ons. And its publishing abilities will redefine the types of content and media published in on-line environments.

“eWorld’s platform will facilitate product distribution and the collection of revenue,” he explains. “We see a major marketing opportunity, as well as a direct channel of distribution, available to developers who use the network.” In other words, eWorld will

News From Apple Business Systems

Getting Started With Netware 4.1 NLM Development

By Robert Patrick

In April, Apple Computer, Inc., and Novell announced the development of a version of Novell NetWare that will run in native mode on PowerPC processor-based Workgroup Servers. This server software, coupled with Apple PowerPC processor-based hardware, will allow developers to create a new class of NetWare applications that support Macintosh, DOS, Windows, Windows NT, and OS/2 client systems.

With the creation of this new, yet standard, server platform, Apple will encourage development of client-server and server applications that specifically exploit the PowerPC performance advantage. A software development kit (SDK) is being readied to help your software development, testing, and debugging efforts.

While the Novell NetWare development environment is well established for the Intel platform, there remain a wealth of opportunities for NetWare development on the Macintosh platform. Developers will soon be provided with a complete set of tools for creating, testing, and debugging PowerPC NetWare Loadable Modules (NLM) compatible with the NetWare 4.1 release. If you’re already developing NetWare 4.x NLMs, you’ll be able to use these tools to recompile and link existing NLMs for the NetWare on PowerPC platform. This is because NetWare for PowerPC processor-based Macintosh computers will have the same NetWare application programming interfaces (APIs) used by other platforms. Once your code is compatible with any NetWare 4.1 platform, your code will be portable to other NetWare 4.1 platforms, allowing you to leverage your development across several platforms and potentially increase your revenue.

You’ll need a solid understanding of the NetWare operating system and architecture to create useful NLMs. NetWare’s design allows good performance for a variable workload and provides efficient service to multiple client systems, but its architecture differs significantly from that of a general-purpose operating system, such as UNIX. When creating an NLM, you must consider and at times exploit these differences. Some of the differences between NetWare and a general-purpose operating system are listed in the table on page 9.

From a user perspective, the NetWare operating system provides a number of services for clients connected to the network. The standard NetWare operating system provides clients with file, print, directory, security, and accounting services. Add-on NLMs from Novell and other third-parties also provide premium services, such as mainframe connectivity or digital video services. With the creation of a PowerPC processor-based NetWare platform, you’ll have new and exciting opportunities for creating and marketing NetWare NLMs that customize Apple’s product and offer additional functionality to clients or simplify administration of a NetWare server.

NLMs consist of code blocks, or threads, built to run on a file server with the NetWare operating system. Specialized code blocks allow you to set up and check the NetWare environment and the server status. NLMs are loaded or unloaded from the server while the server is still running. Once they’re loaded, NLMs actually become part of the NetWare operating system, which means that they can directly access NetWare services without an additional layer of API overhead. This core set of NetWare services or APIs are referred to as the NLM C Interface, which is equivalent to the Macintosh Toolbox.

become another channel for selling your products.

Starting this summer, Apple will include eWorld for Macintosh with all the computers it ships within the United States. Current Macintosh users can also get the eWorld for Macintosh software by calling 800-775-4556. Newton-Mail, an eWorld messaging service, is shipping on every Newton from Apple and its licensees. Each eWorld account, password, and mailbox is the same across eWorld for Macintosh, Newton-Mail, and subsequent versions (including eWorld for Windows).

eWorld will be available worldwide by the end of 1994, with Japanese, French, and German

versions on target for early 1995. You can find more information on eWorld, as well as a HyperCard Vendor Booth application, on AppleLink (path—AppleLink Helpdesk:Publishing in eWorld).

System 7.5 to Debut This Fall

The next version of the Macintosh operating system, System 7.5, will be publicly shown by Apple Computer, Inc., for the first time at the Boston Macworld Expo on August 2–5 and released worldwide to customers shortly after that.

System 7.5, which will run on both 680x0 and PowerPC processor-based Macintosh computers, brings to market a great many attributes requested by customers. Apple is releasing System 7.5 to enhance its competitive position, and to extend the Macintosh platform's leadership in ease of use.

In the words of David Nagel, general manager of the company's AppleSoft division, "With Macintosh System 7.5, we greatly extend our leadership in providing people with systems that are simple to use, easy to upgrade, and bring immediate value to customers. As our competitors struggle to bring to market hard-

ware and software systems that attempt to emulate the Macintosh of a decade ago, we continue to raise the bar to a new level—and to deliver technologies that are focused first and foremost on solving people's real-world work needs."

By adopting system 7.5 technologies—which include AppleScript, Apple Guide, PowerTalk, and QuickDraw GX—you can differentiate your products and help bolster the Macintosh platform's technology lead. To encourage you to do so, Apple will be offering you comarketing opportunities, which *Apple Directions* will tell you about as soon as details are available.

Feature	NetWare	General purpose
Preemptive scheduler	No	Yes
HW memory protection	Optional	Yes
Kernel and user-level boundary	Optional	Yes
Threads	Yes	Optional
Context switching overhead	Small	Large
Address binding	Load-time	Link-time

Differences between NetWare and general-purpose operating systems.

The NetWare NLM C Interface provides the following services:

- network security, management, and accounting
- high- and low-level I/O
- string and file manipulations
- memory allocation
- thread control, synchronization, and communication
- data conversion
- mathematical calculations
- screen management

The development steps for creating a PowerPC NLM are relatively straightforward:

1. Set up development and testing environments.
2. Decide on the architecture of your NLM and divide its work into appropriately grouped threads.
3. Write the NLM code, using the NLM C Interface and keeping in mind the differences between the NetWare OS and a general-purpose OS (for example, Preemption and Threads).
4. Write a definition file that the NLM linker will use to identify start-

up, main, and exit procedures.

5. Compile and link your NLM. Copy it to the NetWare server.

6. Run and test the NLM on a PowerPC processor-based Workgroup Server.

Today, Apple is cooperating with the creation of three development environments that can utilize PowerPC processor-based NLMs. The first two are similar to today's Power Macintosh development environments; they're based on IBM's RS/6000 systems or Macintosh computers with MPW. The third PowerPC NLM environment uses the Cygnus C Compiler in conjunction with Novell's UnixWare software.

For many current Macintosh developers, the MPW solution will be the easiest method for exploring NLM development opportunities. The foundation of the development tools is the Macintosh on RISC SDK, which you're probably familiar with. Its compiler will generate the code necessary for the new NLM linker to create your executable NLM. This NLM code can then be moved onto the NetWare server for testing and debugging.

For a copy of the Apple/Novell White Paper describing Apple's directions with NetWare and the joint effort between Apple and Novell, contact the Apple Developer Hotline at 408-974-4897, or send an AppleLink message to DEVSUPPORT. For detailed information about NetWare NLM development, you can refer to the Novell NetWare NLM Development and Tools Overview, part of the NetWare NLM SDK, which is available by calling 800-NETWARE from U.S. locations. You can register your interest in NetWare or the NLM SDK by sending an AppleLink message to NWDEV. ♣

Robert Patrick (AppleLink: PATRICK.R) is an evangelist in Apple Business Systems Servers and Services Group.

The release makes your life easier, because a variety of technologies that were previously available as separate products are all included standard with the Macintosh operating system. If you build the new capabilities into your application, you no longer have to license the technologies separately.

System 7.5 adds more than 50 new features and technologies to the Macintosh operating system. The following are some of the new features/technologies, listed according to the category under which they fall:

- *active assistance*: Apple Guide, Apple's advanced new electronic assistance architecture; the scriptable Finder, the first version of the Finder that users can customize and automate using AppleScript
- *compatibility*: Macintosh PC Exchange, which lets the Macintosh computer open Windows and DOS files; Macintosh Easy Open, which users employ to open Macintosh documents without the applications that created the documents; the MacTCP client, which gives users access to UNIX-based networks,

including the Internet.

- *collaboration*: PowerTalk, which provides collaborative services that allow users to send electronic mail, share files, and digitally "sign" and forward documents from within an application.

- *productivity*: Macintosh Drag and Drop, which makes it easier than before for users to move data between documents; a hierarchical Apple menu to make it easier for users to access their files and applications.

- *advanced graphics*: QuickDraw GX, Apple's new print and

imaging architecture that offers easier printing, color management, typography, and document portability.

For more information on System 7.5, see "Active Assistance in the Macintosh Interface" and "System 7.5" in the June 1994 issue of *Apple Directions*. (If you're missing your back issue, you can find the files on Applelink [path—Developer Support:Developer Services:Periodicals] or on the July 1994 Developer CD in the What's New folder.) ♣

More Than 130 Native Power Macintosh Applications Currently Shipping

By June 23—barely 100 days after the introduction of the Power Macintosh computer—more than 130 Power Macintosh applications were shipping throughout the world. Native applications have been recompiled to take advantage of the Power Macintosh computers' two- to six-times performance boost over previous 68040-based Macintosh models.

By fall 1994, Apple Computer, Inc., expects hundreds more to join the growing roster of new native Power Macintosh applications. Worldwide developer support for Power Macintosh remains strong; for example, over 22 native applications have been developed in Japan alone. If you haven't begun porting your applications to native PowerPC mode, you'll want to consider doing so, soon.

The following is a list of currently shipping native applications.

Note: This list is provided for informational purposes only; it does not imply an endorsement of any of the products by Apple Computer, Inc.

Developer	Product	Developer	Product
About Software	PM Pro for Mac 2.2	Aldus	PageMaker
About Software	5PM Term for AS/400 2.2	Aldus CoSA	After Effects
About Software	5PM Term for Mac 2.2	ALSOFT	Atlas 1.5.05
About Software	5PM Term for VAX & UNIX 2.2	ALSOFT	Geo Concept
Absoft Corporation	Fortran 77 SDK	Apple	PhotoFlash
Access Privilege	EasyTransfer 3.1	Artifice, Inc.	DesignWorkshop
ACI U.S.	4D Compiler	Artwork Systems N.V.	ArtPro 1.2
ACI	Object Master	Ashlar Incorporated	Vellum
ACI	Object Master Universal 2.5	AS-PLUS B.V.AS-PLUS	Bankafschriften
Adobe Systems	CSPI	Atlas Software B.V.	PS-Mail 2.1
Adobe Systems	Illustrator	auto • des • sys, Inc.	Form • Z Pro
Adobe Systems	Photoshop Plug-in	Baltic Business Systems	MacHansa Acct'g II 2.0 Client
AdventScenery	Animator	B.E.M.E. R & D	ALIX Color
Aetis—Protections	Logicielles Copy Protection	Brossco Systems Oy	Voyant 2.0
Agfa-Gevaert N.V.	FotoLook drivers	Bungie Software	Pathways into Darkness
Aldus	Freehand	Canto Software GmbH	Cumulus

Developer	Product
Casady & Greene	Conflict Catcher II 2.1
Casady & Greene	Spaceway 2000
Central Point Software, Inc.	MacTools 3.0
Charles River Analytics, Inc.	Open Sesame!
Claris Corporation	ClarisImpact
Claris Corporation	ClarisWorks
Conley Corporation	Conley SoftRAID
Dantz Development Corp.	Retrospect
Data Description, Inc.	Data Desk
DataPak	PAIGE
DeltaPoint	DeltaGraph Pro 3
Diehl Graphsoft	MiniCAD
Domark Flying	Nightmares
Dunaway Systems	Signalize 2.6
Dunaway Systems B.V.	PostScript™ Interpreter
Dunaway Systems B.V.	Remote Font & Clip Art
Dunaway Systems B.V.	Scanning&Vektorizing 2.3
Dunaway Systems B.V.	Spooler 1.2
Emergent Behavior	QuickApp
Engineered Software	PowerCADD
Ergosoft	EGWord
ExTools	Shade III 1.1
ExTools	Shade III Light
FIT Software	Full Contact
FractalDesign	Painter 2.0
Frame	FrameMaker
FWB, Inc.	CD-ROM ToolKit
FWB, Inc.	Hard Disk ToolKit
Gibbs and Associates	Virtual Gibbs
Graphisoft, Ltd.	ArchiCAD
Gryphon Software	Morph!
GTPS/GRAFTEK	Ultimage/Pro
Hash, Inc.	Animation Master
Hi Resolution Ltd.	MacPrefect Remote 1.0.1
Hi Resolution Ltd.	MacVisa 1.1
Hi Resolution Ltd.	Mac≈ Bac 1.1
Insignia	SoftWindows™ 286
Interstudio	Domus.CAD 71
Interstudio	flex • plan 1.0
Interstudio	Nonio C 5.0
ITEDO Software GmbH	IsoDraw 2.6
Jasik Designs	MacNosy
Just Systems	ATOK881.3
LEC Language Engineering	LogoVista E to J
Macro Educational Systems	SASI (elementary school version)
Macromedia, Inc.	MacroModel
Maris Multimedia Ltd.	Redshift
Masque Publishing (Ant Software)	Solitaire Antics
MedImage	MedView
Metrowerks	CodeWarrior

Developer	Product
Microland	Le serveur maestria (accounting)
MicroMacro Ltd.	MicroGuard ADB
	Copy Protection
Mimaki Engineering	Vector Link
Neon Software	LAN Surveyor 1.1
Neon Software	NetMinder Ethernet 3.1
Now Software	Now Contact
Orange Micro, Inc.	OrangePC
ORKIS	ImageBasePro 2.5
Pole Position Software GmbH	Mac DCF77
Radius, Inc.	LeMansGT
Radius, Inc.	PrecisionColor 8XJ
Radius, Inc.	PrecisionColor 24X
Radius, Inc.	PrecisionColor Pro 24x
Radius, Inc.	PrecisionColor Pro 24XK
Radius, Inc.	PrecisionColor Pro 24XP
Radius, Inc.	Studio Array 5
RasterOps, Inc.	MoviePack
ROSE	ROSE SongBoard 2.0
Route 66 Geo Info Systems	AtomicTime
Route 66 Geo Info Systems	ROUTE 66 1.2.0 (SignPost 1.2)
Scitex America, Inc.	Full Auto Frame
Segue Software	QA Partner
Société ABVENT	Atlantis
Société ABVENT	Zoom
SofTeam	MacSign 4.0
SofTeam	Punto 1.6
Softing S.r.l	CADSap 4.0
SOFT Technologies	Simulateur de conduite 1.2
Software Architects	Formatter/Drivers
Specular	Infini-D
Spider Island Software	Telefinder Group Edition 3.3
Star Division Corp.	StarView 2.1
Star Division Corp.	Starwriter
Strata, Inc.	StudioPro
System Clinic	DTP603 1.0J
Top Conseil Informatique (TCI)	Top Writer Pro
Trio Systems Europe	C-Index Pro 1.0
TrueVD Software	Live on RISC
Userland	Frontier 3.0
VAMP	McCAD Trailblazer
Vicom Technology Ltd.	VICOM MultiTerm
Vicom Technology Ltd.	VICOM Pro SDK 5.0
Vicom Technology Ltd.	VICOM RunTime
VideoFusion	QuickFLIX
VideoFusion	VideoFusion
Virtus	Virtus VR
Wilkensens SCOOP	SCOOP Archive 1.1
Wolfram Research	Mathematica
WordPerfect	WordPerfect ♣

Technology

Inside This Section

PCI Cards for the Second Generation of Power Macintosh	13
Developer University Schedule	15
Human Interface: Headhunters and Multimedia	16

Technical Support News

Developer Note Describes Newly Released Macintosh and Powerbook Computers

This month's new Macintosh Developer Note Number 10 (APDA #R0568LL/A) provides technical information about three systems that were introduced by Apple Computer, Inc., just as this issue of *Apple Directions* went to press: the PowerBook 150, the Macintosh LC 630, and the Macintosh Quadra 630 computers. These are the first Apple computers to use internal Integrated Drive Electronics (IDE) hard disk drives. All three systems also support external SCSI drives.

Most applications transfer data to and from the hard disk through the File Manager or the Printing Manager and don't require any changes to work with the new hard disk drives. Applications that perform low-level services, such as disk formatting and partitioning, will need to use the new ATA Manager and IDE control calls that have been added to the Device Manager. The added calls and the ATA Manager are documented in the developer note.

The PowerBook 150 computer is a new all-in-one notebook computer that is now Apple's most affordable PowerBook model.

please turn to page 18

CD Highlights

Tool Chest Edition, August 1994

Some of you may remember last month's column, in which I mentioned system software that didn't fit on last month's disc. I promised, instead, to put it on this month's Developer CD. Well, this information was, as we would say here at Apple Computer, Inc., "less than optimally correct." I decided that you would probably rather have the voluminous goodies constituting this month's installment in the Developer CD Series. This month's "goodies" include six new *Inside Macintosh* volumes (featuring the long-awaited *Imaging With QuickDraw*), QuickTime 2.0, the *Dylan Interim Reference*, an alpha version of OpenDoc, a big pile of sample code collected by heroic DTS engineer Martin Minow, and much, much more.

So, in addition to an updated Universal Interfaces folder, here's what's new on this month's disc.



ABS Technical Notes Update

This folder contains new and revised technical notes for AppleSearch, AppleTalk Remote Access, and Apple Workgroup Servers.

AEGizmos 1.3.4

AEGizmos consists of four libraries that provide alternate ways of building and reading Apple events and Apple event

descriptors. They are faster, more memory efficient, and often easier to use than the normal Apple event application programming interface (API).

For many common tasks these libraries have clear advantages over the regular Apple event API, including the following:

- They run much faster when you build complex descriptors.
- They use less heap space.
- They make your code smaller—and easier to write—by requiring fewer calls. (An example is AEBuild, which can replace an entire page of code with one function call.)
- AEPrint, which displays an Apple event or descriptor in human-readable form, can be very helpful in debugging.

AOCE Sample Code

This folder contains several samples showing how to use the components of AOCE/PowerTalk 1.0.

Convert•Projects 1.0b5

Convert•Projects is a utility that reads a Think C or Think Pascal project and produces an equivalent (or nearly so) CodeWarrior project. This is particularly useful for large projects, in which manually adding and segmenting the project would be tiresome or impractical. See the document About Convert•Projects for details.

Convert•Projects is not a source-code converter. If your code uses nonportable constructs that aren't supported by the CodeWarrior compilers, you will need to manually change your code.

Note: This is not an Apple product. It is provided on an "as" is basis. Apple is not responsible for any problems you may encounter in its use.

please turn to page 18

PCI Cards for the Second Generation of Power Macintosh

By George Towner

With the first generation of Power Macintosh computers off and running, Apple Computer, Inc., is getting ready to introduce the second generation during the first half of 1995. These new machines, covering the entire market range of prices and configurations, will present unique opportunities for developers of expansion cards and device drivers. If you design Macintosh plug-in cards, or software that controls or uses plug-in cards, now is the time to stake your claim in this important new technology.

The second generation of Power Macintosh will feature a wide-open system architecture. Apple is replacing NuBus™ technology with the PCI bus, a growing standard in the DOS/Windows world. Apple is also adopting Open Firmware for the system startup process, so Macintosh computers will be able to run non-Macintosh operating systems.

Finally, Apple is taking Macintosh device drivers “native,” to give them PowerPC performance.

What does this mean for you? First, your Macintosh expansion cards will get a big performance boost by using native drivers and a bus that’s more than three times as fast as NuBus. Second, with a little extra effort you can design cards that will work in DOS, Windows, OS/2, and UNIX machines as well as in Macintosh computers. Third, you can save money by using standard PCI-compatible chips in your design. At the 1994 Worldwide Developers Conference, more than 400 Macintosh hardware developers indicated they were planning PCI based products for the second generation of Power Macintosh. We think it’s a big win for everybody—developers, users, and Apple.

Get On the PCI Bus

PCI (Peripheral Component Interconnect) is the emerging bus standard for I/O expansion cards

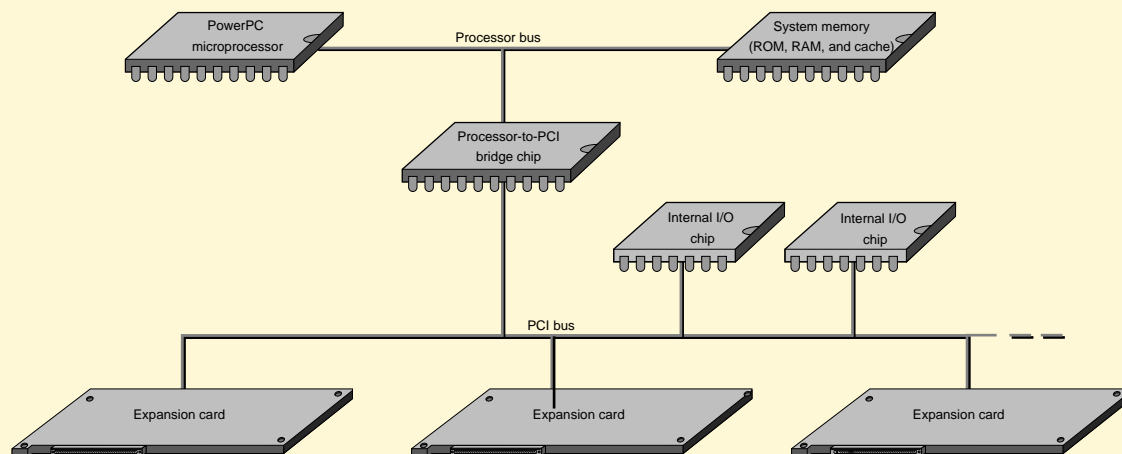
in the personal computer industry. Defined by the PCI Local Bus Specification (see the text box “Technical Information About PCI”), the standard covers bus protocols, card mechanicals and pinouts, voltages and signals, and on-card expansion ROM contents. The PCI bus connects to the PowerPC bus through a bridge chip, and serves both main logic board I/O and plug-in cards (see the figure “PCI Bus Architecture” on this page). The bridge chip controls the bus independently of operating system software. Besides being adopted by Apple, PCI is rapidly becoming the I/O bus of choice in the DOS-compatible and UNIX worlds.

Apple’s implementation of PCI conforms strictly to the PCI standard. With the necessary drivers, your Macintosh PCI cards can work with PCI-equipped DOS or UNIX machines as well, expanding your potential market seven-fold! At the same time, your product costs will go down. Most chip manufacturers are producing

PCI-compatible designs, which are already finding a mass market and are dropping in price. You get cheaper components and a lower investment in custom chips.

Besides giving you a bigger market and lower costs, PCI delivers higher performance. It’s 32 bits wide (expandable to 64) and runs up to 33 MHz, providing a top throughput of 132 MB per second. That’s more than three times as fast as NuBus. PCI has enough capacity to pump full screens of real-time video in 24-bit color, opening up new possibilities for video and multimedia cards. But the advantages of PCI cover the whole range of Macintosh expansion capabilities—current hot areas are high-speed networking, graphics acceleration and buffering, compression/decompression, and high-performance SCSI devices. PCI cards can also use the Macintosh DAV interface to speed up their processing of video and sound data (see the text box “The Macintosh DAV Advantage”).

please turn to page 15



PCI bus architecture. The PCI bus connects to the PowerPC processor bus by a bridge chip located on the main logic board. The PCI bus serves I/O chips on the main logic board (such as serial port controllers) as well as plug-in expansion cards.

What about NuBus? Third parties will offer NuBus-to-PCI bridge boxes, so Power Macintosh owners can still use their existing NuBus cards. But starting next year, the PCI bus will replace NuBus in all new Macintosh desktop models.

PCI and Open Firmware

I said earlier that the PCI bus works independently of operating systems and lets the same card plug into Macintosh, DOS, Windows, OS/2, and UNIX environments. The second generation of Power Macintosh carries this idea a step further by adopting the Open Firmware startup

process.

Open Firmware, originally based on Sun Microsystems OpenBoot, is defined by the new IEEE standard 1275-1994 (see the text box "Technical Information About PCI"). It's a startup and system configuration process, controlled by the Power Macintosh ROM, that takes place before any operating system is loaded. During Open Firmware startup, the ROM explores the computer's PCI bus architecture, locating all the system's I/O devices and constructing a data structure called a "device tree." Code in the expansion ROM of each PCI card registers the

name, location, and technical characteristics of the hardware that the card controls. Each card also stores its drivers in the device tree (see "Drivers Go Native," later in this article).

After constructing the device tree, the Open Firmware startup process searches for an operating system—in ROM, on a hard disk, CD-ROM, or floppy disk, or even over a network. If it finds more than one operating system, Open Firmware may display a dialog box that asks the user to choose one of them. Finally, Open Firmware loads an operating system into memory and gives it processor control.

The operating system accesses the standardized device tree information that Open Firmware left behind and translates it into its own structure of system configuration data.

In implementing Open Firmware, Apple conforms to the IEEE standard, which forms a compatibility bridge between peripheral devices and operating systems. Combined with the PCI bus, Open Firmware gives Power Macintosh users maximum versatility in choosing I/O configurations and system software. Open Firmware compatibility also gives expansion cards the largest possible market and guarantees that your card will work with future versions of the Macintosh operating system.

So your PCI expansion card for Power Macintosh will not only work in other brands of computers—it will work in Macintosh computers that run non-Macintosh PowerPC operating systems. Plug and Play, which Apple has been delivering for years, is looking more like Mix and Match—between PCI and Open Firmware, users will be able to mix and match peripheral cards and operating systems, choosing the products they want to use from multiple vendors.

Drivers Go Native

Your PCI expansion card for the second generation of Power Macintosh computers will normally include an expansion ROM chip containing all the software needed to operate the card. Although it is possible to load support software from a disk or from another source, putting it on board will give your card maximum versatility. The Open Firmware startup process will download your software and make it available to the operating system.

A vital part of your support software will be a set of drivers—one driver for each operating

Technical Information About PCI

Interested in detailed technical information about PCI cards for Power Macintosh? Here are some documents to get you started.

- Apple distributes a technical manual, *Designing PCI Cards and Drivers for Power Macintosh Computers*, plus a disk of tools and sample code, free to serious PCI card developers. To request a copy, send an AppleLink message to APPLE.PCI.
- The essential PCI standard documents for designers of Macintosh-compatible PCI cards are *PCI Local Bus Specification, Revision 2.0* and *PCI Bus Binding to IEEE 1275-1994*. These two documents are available from

PCI Special Interest Group
M/S HF3-15A
5200 N.E. Elam Young Parkway
Hillsboro, OR 97124-6497
Phone: 800-433-5177 (U.S.)
503-797-4207 (International)
Fax: 503-234-6762

- The essential IEEE document for designers of Macintosh-compatible PCI card firmware is *1275-1994 Standard for Boot (Initialization, Configuration) Firmware*, IEEE part number DS02683. You can order it from

IEEE Standards Department
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331
Phone: 800-678-4333

- The American National Standards Institute has prepared a standard called *ANSI/IEEE X3.215-199x Programming Languages—Forth*. It is a useful reference for the Forth language used in the Open Firmware process. You can contact ANSI at

American National Standards Institute
11 West 42nd Street
New York, NY 10036
Phone: 212-642-4900
Fax: 212-302-1286

- SunSoft Press has issued a book, *Writing FCode Programs*, that provides essential information and sample code listings for programmers designing boot drivers for PCI cards. Its ISBN number is 0-13-107236-6. This book is published by Prentice Hall and is available at most computer bookstores.

The Macintosh DAV Advantage

Apple has already established the DAV (digital audio/video) interface for expansion cards in the Macintosh Quadra AV and Power Macintosh AV models. DAV is a separate main logic board connector, accessible to a plug-in card, that taps into the computer's raw data streams for digital video and 16-bit stereo sound. Your multimedia, video, or sophisticated graphics card gets a big performance boost by accessing these data streams without having to pass them through the I/O bus.

The second generation of Power Macintosh logic boards will retain the DAV connector. Combined with the fast PCI bus, DAV gives expansion cards a new range of high-end processing opportunities. For example, a card with compression/decompression features can achieve spectacular throughput by accessing raw video or sound through the DAV connector, using the PCI bus only for compressed data. DAV handles most of the bulk data traffic, leaving PCI free for other tasks.

system that might use your card. The extra effort you put into supplying a range of drivers in your card's expansion ROM will pay off by opening the market for your product to the PCI-compatible DOS, Windows, and UNIX worlds and beyond.

Starting with the second generation of Power Macintosh computers, drivers for the Macintosh operating system must be

written in native PowerPC code. Don't worry—Apple supplies tools and sample sources in C to help you out, and the resulting code will run fast. To help you get your native drivers going, Apple has streamlined the call interface with a single entry point and a more versatile way of handling interrupts. Native drivers are now stored in containers compatible with the Code Fragment Manager.

A new set of system routines, called the *Expansion Bus Manager*, gives the drivers you write today maximum compatibility with future versions of the Macintosh system software, including kernel-based systems.

Besides a set of drivers for various operating systems, your card will need a boot driver if Open Firmware uses it during system startup. For example, if your card operates a hard disk drive that may contain an operating system, the expansion ROM on the card must contain a boot driver. Hang onto your chair—in conformance with the IEEE standard, boot drivers are written in FCode, a tokenized form of Forth! But Apple supplies the CForth93 compiler and a Tokenizer tool to do a lot of the work for you, plus sample code.

What You Should Do

Now is the time to get your products started. The second generation of Power Macintosh will be a major, high-volume CPU family. Your expansion card will be most visible to the market if it's available at introduction.

So send an AppleLink message to APPLE.PCI, outlining your plans. Apple will send you technical documentation, tools, and sample code (see the text box "Technical Information About PCI"). Apple will let you know about events such as Developer Kitchens, where you can work with second-generation Power Macintosh computers. Apple can also give you help with technical and marketing issues.

The second generation of Power Macintosh computers represents a unique opportunity. Now you can design Macintosh expansion cards that combine leading-edge performance with compatibility across the entire desktop computer market. We think that's an opportunity you won't want to miss. ♣

George Towner is a writer in Apple's Developer Press, which produces technical publications for the Apple developer community, including Inside Macintosh and developer notes.

Developer University Schedule

If your technical people need "how-to" instruction in all aspects of programming the Macintosh computer, then Apple Computer's Developer University (DU) is the place for them. With course offerings on a broad range of subjects, from PowerPC and OpenDoc to Newton and MacApp, DU will give your engineers the information they need to build Macintosh and Newton software using the latest Apple technologies.

To help you find the courses you need, *Apple Directions* regularly publishes the latest schedule of DU classes. To receive more information, including a catalog and detailed schedule, or to register for a class, contact the DU Registrar by phone at 408-974-4897 (press 2 when you reach the phone tree), by fax at 408-974-0544, or by AppleLink at DEVUNIV. A list of short descriptions and a schedule is also

available on AppleLink (path—Developer Support:Developer Services:Apple Information Resources:Developer Training).

August

- 1 Newton Programming: Essentials
Advanced C++
- 8 Newton Programming: Extended Topics
Advanced C++
Macintosh Programming Fundamentals
- 15 Newton Programming: Communications
Debugging Strategies and Techniques*
PowerPC Boot Camp
- 16 Designing Object-Oriented Frameworks
- 17 Writing and Using Device Drivers
- 22 Apple Events/AppleScript Programming
Object-Oriented Fundamentals
Programming with QuickDraw GX
- 29 Programming MacApp 3.0

September

Technology Review Series:

- 15 Newton Technology Overview

Regular courses:

- 13 Programming With OpenDoc
- 16 Introduction to OpenDoc
- 19 Macintosh Programming Fundamentals
PowerPC BootCamp
Advanced C++
Newton Programming: Essentials
- 26 Newton Programming: Extended Topics
Intermediate Macintosh Application
Programming

**Classes are offered in Portsmouth, New Hampshire. All other classes are offered in Cupertino, California. ♣*

Human Interface

Headhunters and Multimedia

By Pete Bickford

If you're like me, there's nothing in the world more horrible and dull than listening to a person try to explain what his profession is all about and why it's important. So, since this is one of the things I'll be doing in the next few hundred words, I'd like you all to prepare yourselves for having an absolutely dreadful time of it. Still, for the readers whom I've failed to scare off with the preceding message, I feel it's my duty to provide you with an element of foreshadowing and suspense: I'll also explain why countless VCRs across the world are currently blinking "12:00," and I'll make the connection between the efforts of a hapless headhunter and beautiful, unusable multimedia programs.

The Phone Call

So there I was, sitting at my desk reading my e-mail when the telephone rang. The voice on the other end sounded just slightly too cheerful. "Hi! I'm Lamont Cranston from the executive placement firm of Hookem, Ropem, and Brandem. How are you today?"

People buy things all the time that they don't know how to use. They buy programs with features that they'll never access.

At that precise moment, the industrial strength antihistamine I'd been taking for my hay fever decided to kick in. My brain turned to tapioca for a second and I missed the fact that this guy was obviously a headhunter. "Er . . . just fine," I stammered. "What can I do for you?"

"I see here that you're a Human Interface Champion™ for Apple. Tell me, what is it exactly that you do there?"

"Well . . ." I began, fighting to shake off the effects of drug, "it's basically about making sure that our products are easy for people to use."

"That's great!" the cheerful Lamont said. "So you design the keyboards and monitors and stuff?"

"Not really," I replied, starting to figure out who this guy was. "That's really more of an industrial designer's job. Interface designers do some of that, but we mostly work with software."

"Oh, I get it! So you're like an artist, right? You do all the graphics and displays and things like that, huh? Ya know, that's really important stuff right now, what with those gooeys and all. . . ."

His use of "gooey" (GUI)—my least favorite term in the entire world—sobered me up immediately. It was then that I decided my

mission in life for the next five minutes was to tell this yutz exactly what an interface designer did.

"Screen design is a big part of what I do, but human interface designers have to address the bigger issues; we need to worry about the way the screens work—not just how they look. Often, we'll work with visual designers who do the actual graphics that appear in the system. Me, I can hardly draw at all."

"Oh, I get it! So you program the screen displays and menus and things like that!" he declared triumphantly.

My throat clenched. "Well I can program a bit, but really I'm more concerned with making sure the system lets users get their jobs done easily. Interface designers need to start by figuring out what users are trying to accomplish, then help engineers build a system that lets them do it without a lot of hassle. We talk to users, build prototypes, test them with users, then work to do the million-and-one things that go into making the interaction between the user and the system smooth and polished."

"Hmm . . ." he intoned, in a tentative sort of way. "So you're sort of a marketing guy?"

Suddenly, I felt bad for this guy (not to mention feeling pretty useless myself, since I obviously wasn't good for anything, not being a visual designer, programmer, marketer, or anything else). "I guess interface design has a lot to do with marketing," I said, trying to throw him a bone. "But we represent the user—the person who uses the system. Marketing is mostly about customers, trying to figure out how to sell a system to someone. My job is about making sure that the person buying the system can actually use it."

"But aren't they the same? I mean, who's going to buy things they can't use?"

"Lots of folks," I replied sadly. "People buy things they can't really use all the time. If you need proof, are a couple million VCR clocks blinking 12:00 right now."

"Ah!" he said.

The Problem

"Ah, indeed," I thought. "That really is the problem, isn't it?" It took me another few minutes to shake old Lamont, but I won't go into the gory details here. In a way, I'm grateful to him, because our conversation helped me understand that basic problem: People buy things all the time that they don't know how to use. They buy stereos with controls that will never be meaningfully adjusted. They buy programs with features that they'll never access. And they buy VCRs that will blink "12:00" until their teenage kids set it for them. People purchase these things because of factors ranging from the product's looks to its perceived "power" to its price. It's only when they get the darn thing home that they really think about working with it. By then, it's too late.

Or is it? You might say that usability is one of the fastest-growing after-markets today. VCR programming has proved so troublesome

that millions of special remote controls have been sold for the sole purpose of letting people more easily record the shows they want to watch. Similarly, a legion of DOS shells (including the big one: Windows) have been sold to people who were tired of struggling with the cryptic commands required to do basic things with their home computer.

But even now, usability and human interface are concepts that elude a lot of product designers. Maybe the most damaging misconception is that interface designers are overpaid graphic artists, and that interface design is just a matter of making the product look pretty. The following “case study” shows that this just isn’t the case.

Beauty Isn’t Everything

My wife recently picked up a writing assignment in which she had to review close to 200 CD-ROM programs and pick the best 50 or so. I’m an inveterate media junkie, so I gladly watched over her shoulder while she went through disc after disc.

In addition to being impressed by the huge number of titles that have appeared in the past year or two, I was struck by two things: first, that almost all the titles are absolutely gorgeous; second, that many of the most beautiful ones have the worst interfaces.

It used to be that programmers designed the interface, with the usual result being ugly screens crammed with text. Today, multimedia interfaces seem to represent some sort of artist’s revenge. The screens are beautiful, full of marbled backgrounds and tasteful shading. Any text you see is undoubtedly in Tekton, or some similar artful and decorative font, styled just right to coordinate with the overall look.

In putting aesthetics above all else, however, something gets lost: the user. Tekton is a fine font, but it’s murder when you have to read 20 screens of it in an encyclopedia entry. Similarly, those lovely controls are often so artfully blended into the backgrounds (no outlining of controls in black here!) that the user is left not knowing where to click.

Navigation is a nightmare with many of the discs, with no clear path between the various parts of the program. Labeled buttons, or even (unthinkable!) a menu bar would be a big help, but these are hidden as a matter of course so as not to upset the “look” of the program. Many programs require the user to remember to type “Command-Q” in order to quit; and one or two didn’t even let you out then—you had to turn off the machine!

The list goes on: consistency among interface elements is virtually taboo, feedback is sketchy (often the best way to tell if the program has responded to your action is to watch the CD-ROM drive’s access light), and too often the user is taken out of the driver’s seat and left to watch helplessly as the program inflicts some slide show or controller-less QuickTime movie on them for the third time.

Learning From Video Games, Part 2

Last fall, in the October 1993 issue, to be exact, I wrote about how video games were some of the best examples of how to use color, sound, and motion to create a great user experience. Back then, I was talking about how using the same elements tastefully within

more traditional programs could add immensely to the effectiveness of the user interface.

But the programmers of traditional programs aren’t the only ones who should grab a roll of quarters and head down to the video arcade. In particular, multimedia developers should spend a long time researching what factors make for a good game. For instance, a good video game has the following qualities:

- It’s highly interactive. Users spend most of their time doing things rather than just watching the system.
- A good game is immediately approachable, requiring little or no instruction.
- It’s almost infinitely responsive to the user’s actions.
- It’s rich in texture and detail, yet the points of control are obvious.

Video games are a great example of the partnership that needs to exist between marketing, engineering, industrial design, graphic design, and human interface. These are different disciplines, and each needs to do its part. If the game concept is dull, the programming faulty, the graphics substandard, or the play confusing, the project won’t be a success. The same can be said for any software product.

As our products grow more complex. . . it becomes more and more crucial that true usability, not just visual appeal, is part of the product plan.

Make Your Products Usable, Not Appealing

For too long, many in the software industry have gotten away with the notion that all they need to do to sell a product as “friendly” is make it look nice. This is the attitude that blessed the world with sleek, matte-black electronic gear with curiously blinking clocks. Image is important, but it can’t make up for usability. And as our products grow more complex—and our customers grow tired of being burned by buying technology they can’t use—it becomes more and more crucial that true usability, not just visual appeal, is part of the product plan.

*Till next time,
Doc*

Developer Note

continued from page 12

The PowerBook 150 is lighter and less expensive than the PowerBook 145B; it also has a faster (33-MHz) microprocessor and a larger screen. Its flat-panel screen is 9.5 inches (measured diagonally) and has a resolution of 640 by 480 pixels.

The Macintosh LC 630 and Macintosh Quadra 630 computers are entry-level desk-

top models specifically designed for multimedia use. They use the 68LC040 microprocessor running at 66/33 MHz, and a built-in CD-ROM drive, and they offer modular expansion. These new desktop models start with the features of the Macintosh Quadra 605 computer plus a built-in CD-ROM drive, then add plug-in modules: a communications card like the one in the Macintosh LC 575, a TV tuner like the one in the Macintosh TV (but with a clearer picture), and video input and output

like the AV Macintosh models.

We'll go into detail about the PowerBook 150, Macintosh LC 630, and Macintosh Quadra 630 computers in next month's issue; in the meantime, you might want to read up on them in the developer notes on this month's Developer CD. Macintosh Developer Note Number 10 is also available in printed form from APDA. (See page 24 for APDA ordering information.) ♣

CD Highlights

continued from page 12

Developer Notes Update

Included here are developer notes for several new products, including the Macintosh LC 630, Macintosh Quadra 630, and PowerBook 150 computers.

The Macintosh LC 630 and Macintosh Quadra 630 computers are new desktop models featuring the Motorola MC68040 microprocessor and new modular enclosures. They combine the functionality of the Macintosh Quadra 605 computer with the video input capabilities of the Macintosh AV models.

The PowerBook 150 computer is a new all-in-one notebook computer that replaces the PowerBook 145B. Compared to the PowerBook 145B, the PowerBook 150 features a faster microprocessor (33 MHz), a larger, higher-resolution screen, and an internal Integrated Drive Electronics (IDE) hard disk drive.

For a more complete description of the new Developer Notes, see the article on page 12.

Dylan Related

This folder provides information and sample code pertaining to Dylan, the new object-oriented dynamic language (OODL) from Apple. The folder includes the following:

- **Dylan FAQ.** This memo answers frequently asked questions about the Dylan programming language and its implementations. The latest version of this memo is available by anonymous ftp from cambridge.apple.com (in plain ASCII) in the file `/pub/dylan/dylan-faq.txt`. A formatted version of it using RTF, which can be read by Microsoft Word, MacWrite, and many other word processors, can be found in the file `/pub/dylan/dylan-faq.rtf`. Other Dylan docu-

ments are available for ftp from the same directory.

- **Dylan Interim Manual.** This directory contains the June 1994 interim Dylan language reference.
- **MacMarlais.** Marlais is a simple-minded interpreter for a programming language strongly resembling Dylan. It is not intended as a final release, but rather to fill a perceived void where Dylan implementations are concerned. This is a "hackers release," and it's intended as a vehicle for education and experimentation and also to encourage people to port it to different architectures, add features, and fix bugs. Marlais is alpha software and should not be used by people looking for reliability!!! For more information, please read the Dylan FAQ document in the same folder.
- **Thomas 1.1/Gambit 2.0.** This is an experimental implementation of Dylan. Thomas is implemented in Scheme and runs on many Schemes on many platforms. In particular, we include a stand-alone Thomas interpreter for the Macintosh computer based on Gambit, a freeware Scheme for Macintosh (and 680x0 UNIX computers).

Inside Macintosh Update

Inside Macintosh is a collection of books, organized by topic, that describe Macintosh system software. Together, these books provide the essential reference for programmers, designers, and engineers creating applications for the Macintosh family of computers. New volumes this month include *AOCE Application Interfaces*, *AOCE Service Access Modules*, *Imaging With QuickDraw*, *Operating System Utilities*, and *Sound*. Also included are six books in the QuickDraw GX series.

MacCalendar 1.0

MacCalendar is a complete snippet that illustrates how to write a module for the Control Strip Manager that is available with PowerBook systems. Clicking on the MacCalendar icon in the Control Strip displays the current month's calendar. A separate application lets you configure the font, font size, and weekday labels.

Macintosh Easy Open Version 1.1f1

Macintosh Easy Open 1.1 (also known as the Translation Manager) allows users to open documents created by applications they do not have. The developer kit lets you create Macintosh Easy Open—savvy applications as well as translation systems that perform translation.

Macintosh Tech Notes Update

Technical notes are collections of short (and not-so-short) articles dealing with specific development topics. Updated for this month is Tech Note HW 30—Sense Lines.

Macintosh Common Lisp Utilities

This folder includes the latest patches and interface files for users of Macintosh Common Lisp (MCL). This is a maintenance release, with some bug fixes and support for MCL on newer Macintosh platforms, such as the Macintosh Quadra AV computer. If you currently have MCL 2.0, the enclosed files will update you to 2.01. In addition, this folder contains tutorials and discussions of various Lisp-related topics.

Macintosh Drag and Drop Version 1.1.1

Macintosh Drag and Drop is new technolo-

gy from Apple that your application should support. By using Macintosh Drag and Drop, you can easily implement intra-application and interapplication drag and drop in your applications. Your products will have an improved user interface, and your users will be able to manipulate and work with their data faster and more intuitively. Version 1.1.1 corrects a problem with the universal interface file.

Memory Monitor

Memory Monitor is a software utility that provides information about memory usage to a programatically driven environment such as Virtual User (VU), AppleScript, or any Apple event-capable environment. Memory Monitor was created to assist test engineers using an automated test environment in obtaining memory usage information without human intervention. It has been used extensively for gathering RAM footprint information for software products. It can also help locate memory "leaks."

Network SW Installer

Version 1.4.5 ZM

This folder contains the Network Software Installer for updating to AppleTalk version 58.1.4, a localized product for installation on international system software as well as for the U.S. System Software. See the Read Me file for details.

Network Watch (DMZ) Version 1.3

The Network Watch (DMZ) program searches a selected zone for all available AppleTalk entities and displays its findings to the user. The program is supplied in "fat" binary format and runs on all 680x0 processor-based systems and on Power Macintosh computers in native mode.

This sample program provides source code to demonstrate the use and implementation of a number of AppleTalk (and related) services, including Name Binding Protocol (NBP), Zone Information Protocol (ZIP), Echo Protocol, implementation of a socket listener, and AppleTalk Transition Queue management.

Newton Sample Code 1.1

This folder contains Newton Q&A documents (in both DocViewer and Microsoft Word formats); sample code for the Newton MessagePad, Newton MessagePad 100 and 110, and

Sharp ExpertPad; and several articles on Newton development. Each Newton project includes a text file containing the project's source code (for curious programmers who don't have the Newton Toolkit).

OpenDoc

This folder contains everything you need to get started writing OpenDoc part editors and container applications. We encourage you to dive right in and get started writing OpenDoc parts. The documentation and code is at the point where in a day or two you can write an OpenDoc part editor that plugs in with the rest of the sample parts included on this CD. We've included some sample OpenDoc parts in the demo folder, so once you install the needed system software you'll be able to try out OpenDoc A6 sample parts.

PlainTalk Speech Technologies

The Text to Speech folder contains information about giving your programs text-to-speech capability. The Speech Recognition folder contains information about using the new Speech Recognition Toolbox and API in your programs. In the PlainTalk for Power Macintosh folder, you can find golden master versions of the complete PlainTalk 1.2.1 system that ships with the Power Macintosh 6100AV, 7100AV, and 8100AV computers for your reference and use.

PowerPC Dev White Paper

This 30-page white paper, written specifically for Apple developers, provides an overview of the Power Macintosh platform from both a business and a technical development perspective. We encourage you to make and distribute copies of this white paper, provided you distribute them without charge.

QuickDraw GX Info

This folder contains the interface files for QuickDraw GX and an alias to the QuickDraw GX-related *Inside Macintosh* volumes on this disc. The final version of QuickDraw GX is expected to ship in August 1994. The software will be available in System 7.5 on a future System Software CD; the QuickDraw GX Software Development Kit (SDK) will be available through APDA.

QuickTime 2.0

The QuickTime system software extension adds capabilities that let your applications integrate graphics, sound, video, text, music,

and animation into documents. By providing a standard way for all Macintosh programs to control these multimedia elements, QuickTime makes them easier to use.

The QuickTime Musical Instruments file contains over 30 different musical instrument samples that can be used in QuickTime movies.

The QuickTime PowerPlug extension provides native Power Macintosh versions of some portions of QuickTime. Using the QuickTime PowerPlug extension will dramatically improve movie playback.

In the new Scrapbook, you can cut, copy, paste, and play QuickTime movies. Larger images will shrink to fit so that you can see the entire image in the Scrapbook window. This Scrapbook is otherwise identical to the standard Macintosh Scrapbook.

ResEdit Version 2.1.2

ResEdit is Apple's standard direct-manipulation tool for creating and editing resources. Version 2.1.2 fixes serious bugs from the previous version (2.1.1) that are reported and reproducible. Several less-severe bugs have also been fixed.

SCSI Samples f

This folder contains several snippets that illustrate the original and asynchronous SCSI Managers, including a working SCSI device driver. All samples work under the Asynchronous SCSI Manager (SCSI Manager 4.3); most also work under the original SCSI Manager (documented in *Inside Macintosh, Volume IV*). They've all been compiled under several C environments (MPW, Think C, MetroWerks).

Coming Next Month

Next month we'll bring you a small redesign based on feedback from the Worldwide Developers Conference and a Grand Unified Theory of interface files. Same hack time, same hack channel.

Alex Dosher
Developer CD Leader

Business & Marketing

Market Research Monthly

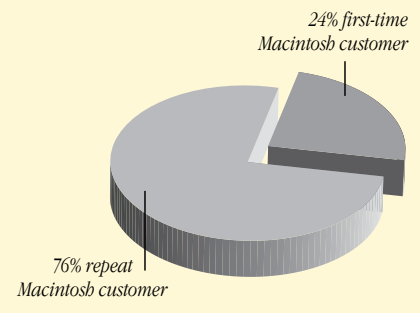
Macintosh Computers Attract New Users

The \$64,000 question for Apple Computer, Inc., is whether it will succeed at increasing market share with its Power Macintosh computers, operating system advances, and other tactics. The company's strategy is aimed at dramatically increasing the Macintosh computer's share of the worldwide personal computer market, thereby increasing the number of potential customers for your products and services, as well.

We don't yet have data on how the Power Macintosh release is affecting Apple's market share, and whether, in fact, the RISC-based Macintosh computers are attracting new customers. However, Apple recently completed a study of customers in the United States who bought Macintosh computers between November 1993 and February 1994—just prior to the Power Macintosh introduction. The study shows that even without the addition of Power Macintosh systems to its product line, Apple was attracting users who were new to the platform.

According to the study, conducted by

First Time vs. Repeat Macintosh Customers



Apple's Customer Research group, fully one fourth—24 percent—of Macintosh customers during those four months were new to the Macintosh computer; that is, they had not previously used a Macintosh computer regularly.

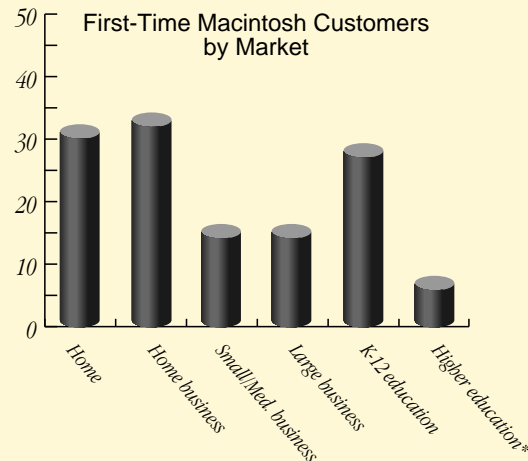
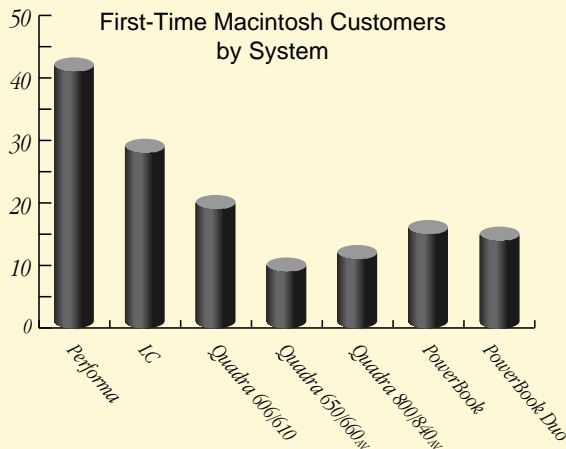
Most of those new Macintosh customers bought low-end systems, either Macintosh Performa or LC computers, and came from the home, home business, and K-12 educational

segments. For complete data, see the graphs on this page.

During the first half of this study, Apple was experiencing record Macintosh unit shipments. During the first fiscal quarter of 1994 (October through December, 1993), Apple sold more than 1 million Macintosh systems worldwide, the first time Macintosh sales reached seven figures in one quarter. Macintosh shipments increased 9 percent in Q2 (January through March) of 1994 over Q2 of 1993, and revenues were \$2.08 billion for the quarter—5 percent higher than the same quarter the year before.

We think this suggests that Apple's market-share strategy began to pay off even before the hot new PowerPC processor-based Macintosh computers were available. Given that, it's intriguing to think about how the Power Macintosh computers will affect the personal computer market; as soon as Apple is prepared to release its Power Macintosh customer data, we'll report it in an upcoming issue of *Apple Directions*. ♣

First-Time Macintosh Customers



*Actual percentages for higher education market are approximate.

Source: Apple Computer, Inc.

Marketing Feature

Getting Maximum Impact From Product Reviews

A Practical Guide for Creating a Reviews Campaign

By John Pearce, MediaMap

Although the product review is only one of many kinds of product press coverage, it is probably the single most important element of a product publicity program. From the standpoint of getting the right kind of information to the right people—that is, the chief buying influencers for your products—the impact of the product review can't be equaled. Because of the function, content, and readership of reviews, their impact dwarfs that of news stories, flattering profiles of your company executives, and corporate image pieces (although those types of stories certainly have their place).

Reviews hit the bull's-eye of the market for a specific product, the real buyers: the small group of individuals that evaluate your product and either recommend it to others in their organizations—or recommend your competition. When you look at it that way, you see that all coverage is not created equal and that product reviews can have an impact that's completely disproportionate to the number of column inches of ink you get, or even the number of readers you reach.

For example, at our company one person makes the product purchase decisions for all our requirements. He doesn't read news stories; he thinks they're fluff, and they're not what he's looking for. He may glance at trend stories, but he does his own thinking about what products will meet our needs. And he

completely ignores the personality profiles and such, because he doesn't care if someone in the industry has a colorful persona or not.

What he does care about is whether a product will meet our needs. And most often he gets that information from product reviews. He's only one reader, but he buys for 20 people. He's your bull's-eye.

Where Are the Opportunities?

There are literally hundreds of review opportunities for computer-related products, not only in our own industry's publications but also in other media. In the United States, roughly 70 publications dominate the computer trade press, and many of them run reviews. But although many developers limit their thinking to the trade press, there are scores of opportunities beyond the obvious ones.

- *National press.* One often-neglected media category is the national press, which includes major magazines and newspapers, syndicated columns, radio and TV shows, and wire services. These can be a gold mine. One example is "Ask Dr. John," a column picked up by hundreds of papers of every kind and size across the United States. Those newspapers reach literally millions of readers, many of whom are potential customers for home-use products. Because the column isn't in the computer trade-press mainstream it's easy to overlook, but it's well worth

the time and effort to pursue.

Another personal favorite is the Associated Press syndicated column "CompuBug." CompuBug runs product reviews that reach more than 400 newspapers throughout the United States. One of our software company clients was reviewed in CompuBug; the review was picked up by hundreds of those papers—and the company was deluged with thousands of inquiries. Now, that's cost-effective marketing!

- *Computer-industry newsletters.* These provide even more opportunities for reviews, especially for more-technical or leading-edge products. Newsletters usually have a very small circulation compared with that of consumer publications, but they can make a huge impact. Many are expensive to subscribe to, so the subscribers—who include most of the computer industry's opinion and market leaders—are very likely to pay attention to the newsletter's review opinions.

- *Roundup stories: The ultimate in reviews.* The roundup story is probably the most important kind of product review, because it is a comparison of multiple products in a given category—and usually points out the best. If you come out on top, you'll really make an impact on potential buyers. There is no more powerful way to work with the press to help position your product. But a word of caution: As powerful as roundup stories are, if your product gets savaged or mispositioned, it can be a

severe blow. As we'll see below, avoiding such damage is ordinarily within your control or at least your influence.

Sometimes roundup stories are scheduled and listed on editorial calendars. So be sure to get the editorial calendars for all your key publications. Find out who will write the story by calling the managing editor (or referring to a resource such as MediaMap). Once you've made the right contact and ascertained the direction and theme of the story, it's time to help the editor understand how your product fits into his mission. Comparison tables honestly contrasting and positioning your product versus others in your category can be a good way to get empirical and even subjective information across to a roundup-story editor. In any case, take personal responsibility for ensuring that the editor truly understands what your product is and is not.

Most often, however, roundup stories aren't formally scheduled on editorial calendars. Instead, editors look for hot market segments to cover without being constrained by a calendar planned months in advance. That's a golden opportunity for you to pitch a roundup-story idea, one that is especially welcome when your company isn't making any "news." However, you must proactively find and act on the opportunity.

This is particularly true in the national press—especially the end-user press—because it isn't hooked into the moment-by-

moment news of our trade. The key is to find a way to group your product with others and give the reviewer the appropriate information. In doing so, you help the journalist by providing a good market niche to report on, and you help your company by pitching a good story that positions your product accurately.

How to Attract Reviewers' Attention

At MediaMap we define public relations (including getting product reviews, of course) as "helping the press do its job." To do that, you must understand that job and make your strategy dovetail exactly with it.

One of the biggest mistakes developers make is assuming that getting a publication to review a product works the same as get-

ting an editor to read and run a press release—not true at all. The product review process varies widely from publication to publication, and each one may periodically change its policies and procedures. The key to getting good product reviews is understanding the differences in the product review process and editorial mission between publications.

To succeed, you must understand (and keep up with) the process for each target publication and make sure your efforts complement the publication's review process. Otherwise you'll be very frustrated. You'll send boxes of products into a black hole and wonder why you're not getting reviewed. Most of the time, it's because your approach and methods are on a different wavelength than those of the

publication.

For example, if you don't do your homework for a particular magazine, you may not discover that in a given case the "product review editor" is not an active product reviewer but is instead a front-end person or administrator, someone who receives calls from developers and says, "Yes, thank you very much, please send us your product." The actual reviewer may be sitting on the beach in Malibu with a portable computer.

It's also useful to know such things as whether a publication uses a testing laboratory. Knowing that and understanding the lab's evaluation process is the key to getting a review in such publications. For example, the *Infoworld* review process is unique. *Infoworld* doesn't use a process in which you send a pack-

age to somebody who opens it, looks at the product, and decides what the review will say. Instead, the executive editor of reviews coordinates the activities of five review teams, a testing lab, and a reader survey operation.

Each team is responsible for specific kinds of technologies, and within each team the members have designated responsibilities. The testing center works with the review teams and other editors to determine review goals. The results of the review are combined with results from a survey of 1,000 readers to determine users' hot buttons. Needless to say, simply having a slick pitch on the phone isn't going to dovetail with the entire process.

You also should be familiar with the format and review content of a publication so that you

The High-Tech Marketing Companion **Proven techniques for solving your toughest marketing problems**



In this book by Dee Kiamy and the editors of *Apple Directions*, industry experts and developers describe practical, proven techniques for solving the toughest marketing problems facing your business. You can read about

- how to avoid the ten most common product launch mistakes
- choosing and sticking with the best target market
- getting more and better product reviews
- developing packaging that helps your product stand out on the dealer's shelf

And much more!

To order, call Computer Outfitters at (800) 260-0009 (U.S. and Canada) or (406) 758-8000 (other countries). The book is also available at booksellers throughout the United States and Canada.

can better position your product when you send it for review. Use that information to select the hot buttons you'll press when dealing with that publication.

Your interaction shouldn't be "Gee, won't you review our product?" It should be something more along the lines of "I hope you'll consider our product for review. Since you serve readers in the hog-farming industry, we've put together a little package of information relevant to your readers' interests, including how we stack up against manual hog-accounting techniques. . . ." and so forth.

We did a major survey of the media for a client. Many editors told us that they particularly appreciate receiving a well-thought-out package of supporting materials with review copies of products. Such materials position each product against the other products in the same niche and personalize the product features and benefits for the specific publication and audience. You can also include backgrounders about the technology or your company. This is similar to the press-kit concept, but it's designed specifically for a product reviewer.

Creating the Program

Getting reviews takes planning, foresight, and some degree of commitment. So rather than approaching them helter-skelter and opportunistically, I suggest using a campaign approach. Putting a working system in place and getting the first review is the most arduous part of the process (especially for smaller companies whose resources are stretched) and the most resource-intensive.

As is true with most PR efforts, the most needed resource is usually manpower. But pay that price, and you're in a great position: The incremental effort to get the next review is much smaller, and when you get the machine

going and your fifth, tenth, and twentieth reviews come in, you'll realize the full power and incredible cost effectiveness of a good product review campaign.

Start by making someone directly responsible for the campaign. That person will plan and systematically execute the campaign and do the needed record-keeping and contact work. Then determine how many opportunities you will pursue and in what priority and order. With more than 200 review columns to choose from, you must be selective; base your decision on who your target market is, whom the publication reaches, and the relative impact of each publication. In most markets, it's reasonable to choose 20 to 50 publications, but of course it depends on your goals and resources and how many publications reach your market.

After selecting your targets, you must research each one's review process. As I said before, each process is different—and that difference is crucial to your success. Depending on your resources and time, you may opt to do this research yourself or you can refer to research services. (For example, every 13 weeks we issue a new report that describes the review process of hundreds of publications.)

If you opt to do it yourself, you can contact the publications and find out what their review process is by talking to the managing editor, reviews editor, or even editor-in-chief. This will take time, but you need this knowledge before you shape your strategy.

You also need to create a record-keeping system that tracks every review target on your list. Your system can be as simple as a form for each publication or as complex as a full-blown relational database integrated with your other PR activities, depending on your needs and resources. In any case, here's some of the informa-

tion you need to record:

- For this publication, who is the audience, what is the editorial mission, and what kind of editorial posture does it generally take?
- Who are the product review contacts? What are their names, addresses, phone, and fax numbers? What is each person's role? Who are the primary versus the supporting players?
- What kind of system does the reviewer use? What computer, what operating system, what version, and what media format does the editor need? (Reviewers are not terribly inclined to rave about products they can't use!)
- What's the lead time for reviews in this publication?
- Does the publication honor nondisclosure agreements? If so, under what conditions? What are the implications for any product-announcement schedules you may have?

Your form should also include space for notes about each conversation you have with editorial contact people so you'll know when you've spoken to whom and what they said. A callback date tickler function is also very desirable for tracking your commitments to editors and for reminding you to take needed steps.

With this information in hand, you're almost ready to go. Using this information and your product positioning, tailor your message to each publication's special mission and audience. Then you'll be ready to contact the editors.

Steps to Success

There are several steps to initiating the review campaign, all of which should probably be check boxes or blanks on your record-keeping form for each target publication. You must make the initial contact with the editor, determine whom to ship the product to, follow up to make sure the material was received and is in good working order, and

find out if the reviewer needs additional information and support. If appropriate, send reviewers guidelines or support materials, and test various approaches. Throughout the process, don't forget to keep notes of your conversations and any reactions you get from reviewers.

In addition, be sure the person responsible for your campaign has good in-house support. If the person isn't capable of handling all the responsibilities, identify and brief all the appropriate individuals, especially a technical support expert, a marketing/positioning expert, and an administrator who knows or can ascertain the status of your interaction with each target publication. These in-house players must be reliably available when needed by a reviewer.

Your campaign should also include a systematic evaluation program, a plan for making use of the feedback you get from reviews and reviewers. If you listen, there are many benefits: First, you may head off an unfavorable review before it ever sees the light of print. Second, you may get realistic feedback that will help you tailor your approach with the next reviewer, or revise your support materials or messages. Third, you can regularly report incoming reactions back to marketing and product development so they can make adjustments in their thinking as well. Putting reviewer feedback to good use punctuates the whole process.

Timing Your Campaigns

For a new product, reviews are probably the most important press coverage you could possibly get, and you should plan your review strategy months before the actual announcement. For each target publication, devise a plan that takes into account lead times as well as nondisclosure and embargo policies.

Editors will be candid about their embargo policies (that is, agreements not to release information until a given date), and you should make the agreement explicit if you expect them to hold the review until your announcement date. Although many editors will refuse to sign formal agreements, they almost always honor their commitments; an explicit verbal agreement that you follow up with a letter stating your understanding of the agreement is usually adequate.

Editors who break such agreements are quite uncommon, but misunderstandings can occur if communication is not clear. Remember also that many publications, especially newsweeklies, explicitly do not agree to embargoes; without an explicit agreement, there's no such thing as being "off the record."

The best approach is to handle the timing issue by conducting a preannouncement product tour—anywhere from a few weeks to a few months before the announcement date—featuring nondisclosure presentations to key reviewers, news editors, and analysts (again, make sure that agreements are clear). Such face-to-face interaction helps you build credibility, even in advance of your having any news to announce.

When Not to Solicit Reviews

It's not reasonable to expect a full review about a marginal prod-

uct change. And if you cry wolf by launching an all-out product review campaign around such an upgrade, you run the risk of blowing your credibility for future efforts.

A good alternative for a less substantial upgrade or release is simply to announce it with a press release and "phone-athon" to news editors. Limit your pitch to the two or three key upgrade features that have the greatest market impact. If the upgrade is too minor to make the news pages, you can still get substantial coverage because most publications have a column for product upgrades and minor product announcements.

Dealing With "Unfair" Reviews

Often, developers become frustrated or irritated because they think they've been unfairly reviewed. In spite of how it occasionally seems to an enraged product manager, editors are rarely out to kill products; they're in the business of trying to describe them accurately.

In truth, when a developer feels wronged by a review, three of every four times the responsibility rests with the developer rather than the reviewer. Your responsibility is to make sure the reviewer can't misunderstand the product. A good example is a very high-end application—a power-user product—that gets panned by a reviewer because it wasn't easy to use. In such a case,

the reviewer probably didn't understand the product's positioning. The developer must take the responsibility—and invest the effort it requires—to help the editor understand the product.

But if you feel that you've done all the right things and the review is still negative, squelch the natural desire to give the reviewer a piece of your mind or to take retaliatory action, such as canceling your advertising in that publication. Don't forget that you're in it with the press for the long haul, and act accordingly. If you feel that a reviewer has incompetently reviewed your product, by all means take action, but do so rationally.

Deal with your problem through the appropriate channels; convey your concerns to the people in charge (such as the editor-in-chief) in a reasonable and objective way. Be very specific and concrete about how you feel your product was misrepresented. Also, many publications provide some kind of feedback forum, such as a reviews-response or letters-to-the-editor column. Take advantage of it. And realize that in the longer run your calm approach will work: If editors are incompetent, they won't be around long and you'll still have a good relationship with the rest of the editorial staff. If they are competent and have simply made a mistake, they're quite likely to bend over backward to avoid a similar error in the future.

Another thought: View a less-

than-glowing review as a unique opportunity to acquire useful market feedback. Usually the editors who review products do know what they're doing, and sometimes a product is criticized because it actually has a deficiency. If you consistently get hammered in a certain area, take it as a signal. One of your toughest jobs is telling your executives that perhaps the reviewer is right and that your company should act on the feedback.

Here's a story about a bad review that has a happy ending: A year or so ago, a company got a particularly bad review. Instead of getting huffy, the company examined every criticism, went back into the labs, fixed the problems, and one year later got an editor's choice award from the publication that had panned it before.

Reviews are the driving force behind the development of the market's consensus opinion about your product. If you properly manage the process of getting reviews, the quality and quantity you get will increase dramatically—and hopefully so will your sales, company standing and image, and competitive position. ♣

John Pearce is the founder and CEO of MediaMap. MediaMap, located in Cambridge, Massachusetts, provides PR research and automation systems to client companies in the computer industry.

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*.