The Developer Business Report *June 1995*

AppleDirections

Inside This Issue

Apple News

They're *Heeeeeere. . .*

Power Computing Delivers First Mac OS Compatibles

By the time you read this, Power Computing Corporation, of Milpitas, California, should have delivered the first Mac OS–compatible computers to customers. On April 17, 1995, Power Computing announced its first Mac OS–compatible computers, adding that its first customer shipments would begin on May 1, "ramping to volume shipments in July 1995."

The Power Computing announcement is significant because it represents an early milestone in Apple's licensing policy—the first company to ship a Mac OS–compatible computer. Other companies—most notably, Radius, Inc., Pioneer Electronics Corp., and DayStar Digital, Inc.—have announced Mac OS–compatible computers.

In the second half of 1996, the Mac-compatible market will enter a new phase with the availability of Common Hardware Reference Platform (CHRP) computers, to be built by Apple, IBM, and other vendors. (For details, see "This Changes *Everything*" on page 1 of the December 1994 issue of *Apple Directions*.) This will result in an increased installed base of computers capable of running the Mac OS—and, therefore, a larger market for your Mac OS–compatible products.

Licensing is just one part of Apple's strategy to increase the market share of the Mac OS—a move designed both to help Apple and to

please turn to page 11

Strategy Mosaic

Copland: Technology for Customers' Sakes

By Paul Dreyfus, Apple Directions *staff*

The Goals Behind Apple's Next-Generation Operating System

Whenever Apple Computer, Inc., undertakes a major operating system release, you can assume it's doing so to maintain its technical advantage. Once upon a time, "technical advantage" might have translated to "technology for technology's sake." Today, though, it means "technology for customers' sakes"—that is, making Macintosh computers even more powerful, efficient, and easy to use for all customers, from the least experienced home users to high-end technical users. To put it in crass business terms, Apple revamps the Mac OS in order to draw customers—and developers—to the Apple camp.

All of the above can certainly be said of Copland, which, when it's complete in mid-1996, will be a redesigned, microkernel-based version of the Mac OS, almost fully in "native" PowerPC code. While the competition struggles to build into their systems features that Macintosh customers have enjoyed for years,

please turn to page 4

AppleDirections

Volume 3, Number 6

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
Kris Newby (NEWBY.K)

Associate Editor
Anne Szabla (SZABLA)

Production Editor
Lisa Ferdinandsen (LISAFERD)

Contributors
Ken Bereskin, Peter Bickford, Alex Dosher,
Gary Little

Manager, Developer Press
Dennis Matthews

Manager, Apple Developer Periodicals
Mark Bloomquist

Production Manager
Diane Wilcox

Prepress/Film
Aptos Post

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

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

Hedging Your Investments

When Apple Computer, Inc., introduces radical, ground-breaking new technology it can pose a dilemma. Sure, you think it's great, but it can also feel like a huge risk to suddenly adopt a new way of doing things, especially when the old way seems to work so well.

Such is undoubtedly the case for many of you thinking about OpenDoc and Copland, the latest strategic technologies from Apple. We've been telling you about OpenDoc for some time, urging you to adopt its new component model for software. (Our latest OpenDoc coverage starts on page 15 of this issue.) This month, we start talking about Copland, the next-generation version of the Mac OS (see the Strategy Mosaic on page 1), and, given how important it is to the Mac OS platform, we're going to keep talking about it for the foreseeable future.

You may be thinking that OpenDoc and Copland are great, but you may also feel that adopting them, in their entirety, is just too great a risk for you to think about right now.

There's a way that you can significantly lessen that risk by hedging your investment in future technology—actually, several ways: by adopting AppleScript, Apple Guide, the Thread Manager, and Macintosh Drag and Drop. These are well-established technologies that help make today's Mac OS the most efficient and productive personal computer operating system. Adopting each of them requires an investment on your part, but it's an investment with two potential pay-offs: First, it will help make your product more appealing to today's customers; second, it will prepare your product to take advantage of the Mac OS of the future.

- By adopting AppleScript and making your application Apple event-aware—that is, by making it scriptable—you give your current customers the ability to customize your product so that it works, and interacts with other software, the way they want it to. That's the pay-off today. Tomorrow, your product will be ready to take advantage of active assistance in the Copland release of the Mac OS, and it will be far easier to turn it into an OpenDoc part (or parts). (For more about active assistance, see "Creating the Best User Experience" and the sections that follow in this month's Strategy Mosaic.)
- Apple Guide will make your current product easier to use today, giving your customers access to unprecedented levels of online help. By giving your customers Apple Guide support today, you'll also be ready to adopt the Apple Guide "do-it-for-me" feature of tomorrow that will be an important part of Copland active assistance.
- Using the Thread Manager today, you can build cooperative multitasking into your product today more simply and quickly than ever before, making your customers more efficient with less effort on your part. Tomorrow, you'll be familiar with many of the basic concepts you'll need to know to have your application take advantage of Copland's preemptive multitasking and protected memory space.
- Macintosh Drag and Drop provides a simple, consistent way for users to move content of any kind from one place to another on their desktops. Applications that use it today give customers the benefits of efficient data exchange between documents; tomorrow, the same applications will be more able to interact with OpenDoc software, which has the drag-and-drop capability built into it.

There's no doubt that Apple will continue to trumpet the advantages of OpenDoc and Copland (and other new technologies, as well). If thinking of adopting these technologies requires too large a leap of faith for you, though, think instead of the smaller steps you can take. We don't mean to suggest that these tasks are all you'll need to do to adopt OpenDoc and Copland. However, these smaller steps with today's technologies can, in concert with others, become the giant leap you'll need to take into the future.

Paul Dreyfus
Editor

IndustryWatch: News & Perspective

Fast and Getting Faster

IndustryWatch is our regular compilation of news about events happening outside the Apple Computer, Inc., R&D complex. Each month, we gather the most notable items, which we hope will point you toward new opportunities, help you avoid mistakes, and alert you to key developments in the industry. We're not trying to cover everything in the computer industry; other publications already do that. Instead, we're digging through the news to present you with the most interesting tidbits, ones that translate into ideas you can use.

1994 Macintosh Software Sales Accelerate Faster Than DOS/Windows

The word for Macintosh software sales is *momentum.*

Last year, in North America, Macintosh software sales grew more quickly than combined sales of DOS and Windows software, according to recently released Software Publishers Association (SPA) yer-end 1994 sales data. Macintosh sales grew especially quickly in the last quarter of the year, which included that all-important holiday buying crunch, and in the two fastest growing software categories. The chart "Macintosh Software Sales Outpace Industry in 1994" on this page includes key data from the SPA report.

According to the SPA, North American Macintosh software sales increased 18.7 percent in 1994, to $1.25 billion, three percentage points higher than the overall growth of personal computer software sales in North America last year. Overall, software sales grew 16 percent in 1994, to $7.4 billion from the 1993 total of $6.3 billion.

By comparison, sales of software for PC-compatible systems running DOS or Windows (or both) grew more slowly than the overall industry. Combined DOS/Windows software sales increased only 13.7 percent last year, to just over $6 billion.

Macintosh software sales took off in the fourth quarter of 1994, growing 24 percent over sales in the fourth quarter of 1993; combined DOS/Windows sales increased just under 18 percent over fourth quarter 1993 sales, while the industry as a whole grew 21 percent over the same period.

In the two fastest growing software segments—entertainment and home education—the difference between Macintosh and DOS/Windows sales was even more extreme. In the entertainment category, which overall gained 56 percent in 1994, Macintosh software sales leaped 83 percent, while combined DOS/Windows sales increased 54 percent. In home education, which grew an overall 88 percent in 1994, Macintosh sales more than doubled, growing 111 percent, while combined DOS/Windows sales increased 77 percent.

Implications/Opinions: By itself, this is great news for the Macintosh community. In addition, the data provides yet one more "Damn Good Reason" to develop for the Macintosh platform: Macintosh customers buy more software than DOS/Windows customers. (For 35 other reasons, see page 23 of our April 1995 issue, in the article "Soul of the New Macintosh.") Here's why we draw that conclusion.

According to current data, Macintosh hardware sales are growing at the same rate as computer sales throughout the industry. The SPA data shows that Macintosh software sales are outpacing the industry, while sales of software for Intel-based DOS/Windows machines are growing slightly more slowly than the industry. Putting two and two together,

Apple Directions Online—July

The July issue of *Apple Directions* will be available on AppleLink, the Internet, and eWorld by June 15, at the following locations:

AppleLink: path—Developer Support:Developer Services:Periodicals:Apple Directions.

Internet: select Developer Services and Products at the location www.apple.com.

eWorld: in the Apple area of the Computer Center.

Macintosh Software Sales Outpace Industry in 1994

	Overall*	Macintosh	DOS/Windows
All categories			
'93 sales (000,000s)	$6,329.2	$1,052.2	$5,329.2
'94 sales (000,000s)	$7,381.8	$1,248.9	$6,061.3
'93–'94 % change	+16.6%	+18.7%	+13.7%
Q4 '93 sales (000,000s)	$2,044.8	$321.1	$1,749.9
Q4 '94 sales (000,000s)	$2,477.5	$396.9	$2,063.9
Q4 '93-'94 % change	+21.1%	+23.6%	+17.9%
Entertainment			
'93 sales (000,000s)	$459.3	$51.1	$405.7
'94 sales (000,000s)	$716.2	$93.6	$622.5
'93–'94 % change	+55.9%	+83.1%	+53.4%
Q4 '93 sales (000,000s)	$218.5	$23.6	$193.1
Q4 '94 sales (000,000s)	$358.2	$47.1	$311.1
Q4 '93–'94 % change	+63.9%	+99.6%	+61.1%
Home education			
'93 sales (000,000s)	$277.5	$45.3	$235.9
'94 sales (000,000s)	$522.2	$95.7	$424.9
'93–'94 % change	+88.1%	+111.3%	+80.1%
Q4 '93 sales (000,000s)	$129.1	$23.1	$107.2
Q4 '94 sales (000,000s)	$233.2	$43.4	$189.3
Q4 '93–'94 % change	+80.6%	+87.9%	+76.5%

*Totals include sales of software for other platforms.
Source: Software Publishers Association.

Macintosh customers are currently spending more money on software than their DOS/Windows counterparts.

This does not mean that the Macintosh software market is overtaking the so-called Wintel platform (don't we wish!). But it does mean that overall size is not the only indication of the viability of a market. The Macintosh market is large and growing—and it's growing more quickly than the rest of the market. The way we look at it, some of you Macintosh developers are pulling in strong revenues, especially in categories where there's high consumer demand, namely entertainment and home education software. With the fourth quarter of 1994 indicating that the rate of Macintosh sales growth is only accelerating past the rest of the industry, revenues should only get stronger.

Home Computer Installed Base to Nearly Double by 1997

And speaking of the consumer market: According to a recent report issued by Forrester Research, a consulting firm based in Cambridge, Massachusetts, the number of computers purchased for home use will total about 40 million by 1997. That's almost twice the current installed base of about 23 million home computers, according to Forrester's estimates. Forrester says shipments of home computers will exceed 6 million units this year alone, a 27 percent increase over 1994 shipments. That's at least twice the growth Forrester anticipates for business personal computer sales.

Implications/Opinions: Education and entertainment developers take note, especially if you don't yet sell a native Power Macintosh product (and really, truly especially if you only have products for DOS and Windows machines): This is an opportunity you don't want to miss.

With its recently released PowerPC 603–based Power Macintosh 5200/75 LC computer, priced at under $1,700 (U.S. only) for the education market, Apple has proven its ability to remain the price/performance leader in the consumer computing market. In other words, Apple intends to sell vast numbers of Power Macintosh systems to home users, and we think you'll want to have products ready to sell right along with them. As discussed in the previous item, sales of entertainment and home education software—two of the most popular home software categories—to Macintosh customers are already the hottest, fastest growing spot of the market. If Forrester's research and current buying trends hold true, Macintosh home software sales should positively sizzle for the forseeable future.

Did You Know . . .

. . . that Windows 95 makes only limited use of Unicode, the standard for cross-platform international software development? According to an article in the May 1995 issue of *Microsoft Systems Journal,* only five Windows 95 application programming interfaces (APIs) take Unicode strings. Other string APIs return an error code if they're passed Unicode strings.

And, did you know that "Bob," Microsoft's easy-to-use human interface product for beginning users, doesn't provide Windows customers a scalable introduction to using more advanced system features? In other words, beginning users can learn Bob easily, but once they decide it's time to use full-blown Windows, what they've learned about Bob doesn't really make things any easier for them.

Implications/Opinions: On the other hand, Copland, Apple's forthcoming next-generation version of the Mac OS, provides extensive Unicode support, making it a global operating system—one that can be localized easily. If you write software for Copland, you'll be able to more easily market it simultaneously in geographies that use different writing systems.

Further, the Copland user experience employs scalable human interface features that lead beginning users from a basic, easy-to-use scheme through intermediate levels to the full-blown Mac OS feature set. Along the way, each level trains the user to move to the next level. What approach do you think real-live computer customers are going to prefer?

The Race to the Toll Booth

Eight major U.S. newspaper publishers, including such biggies as Times-Mirror Co., Knight-Ridder, Inc., and Washington Post Co., recently formed a joint venture called *New Century Network* to link their online news services into a single network. New Century Network, which is being funded by the eight newspaper companies, intends to link news stories from nearly 200 different newspapers in a single World Wide Web site within the next three years and charge subscribers who log on to the site.

Implications/Opinions: This will be a great service for news hounds. The sticky wicket, however, is actually charging people to access a Web site. So far, no one has figured out how to build an elegant information superhighway toll booth, even though lots of companies around the world have announced plans to sell services and information over the Internet. Many are trying, and the person who comes up with an easy-to-use, seamless, secure way of charging online customers for Web site access is going to make a bundle. That person might as well be you. ♣

Strategy Mosaic

Copland

continued from page 1

Copland engineers are designing technology and features that you've never seen before, while at the same time providing Macintosh developers and customers with a more powerful, stable platform. For example, I think you and your customers are going to love native system performance with preemptive multitasking, not to mention a completely threaded, much faster Finder, the new human interface, and—get this!—a brand-new standard file dialog box.

The Hype and How to Avoid It

Long before you see real-live Copland code, you can expect to be bombarded by the press and Apple alike on the wonders of the new system. (In fact, it's started already; the latest issues of *Macworld* and *MacUser* both feature cover stories about Copland.) While you're waiting to see Copland for yourself, there are a number of steps you can take today, using existing tools, to get your products ready to work with the new Mac OS. Later in this article, I'll tell you about those steps, so you've got something to distract you from the hype—and, I hope, so you can start getting ready for the next generation of Macintosh computing.

Before you consider those steps, it's important for you to understand the magnitude of the change Copland will represent as well as Apple's goals in developing a major new version of the

Mac OS. (If you're already familiar with those goals, or if you're just impatient, you can skip down to "Here's What You Can Do" on page 9).

Change Equal to (or Greater Than) Power Macintosh Intro

Apple views Copland as the second critical part of the transition to PowerPC, since Copland will be the first native PowerPC release of the Mac OS, with native file system and I/O (input/output) architecture. For you, this amounts to a change on a level equal to, and possibly greater than, the introduction of the first Power Macintosh computers.

Thinking back a little more than a year, you'll probably remember being reassured that the new Power Macintosh systems didn't *require* you to rewrite any code, since they all run your 680x0 software in emulator mode. However, if you wanted to open your product to the rapidly expanding Power Macintosh market and allow your customers to take advantage of native performance, you had to take the steps required to release a native PowerPC version of your product.

Similarly, Apple wants to reassure you that compatibility is a key goal behind the Copland release; most existing products will run under Copland. But there will be many reasons for you to change your products substantially to take advantage of Copland's new features and to work optimally with the new Mac OS. On the other hand—and here's why Copland may present a greater challenge for many of you—Copland will break some existing products, namely system extensions (INITs) and desk accessories (DAs); application features that depend on system extensions won't work under Copland.

According to Vito Salvaggio, the Copland product marketing manager, Apple "thought long and hard about this issue, and we had to make a tough call. Essentially, we had devised an elegant design for Copland, and we didn't want to handicap it with the current patching mechanism for extensions. Also, we've been saying since the release of System 7 that, one day, DAs wouldn't work anymore." For more about how Copland might affect your product, and what you can do about it, see the section "New Run-Time Environment," later in this article.

The release of the first Power Macintosh computers also attracted non-Macintosh developers interested in capitalizing on Macintosh RISC-based performance. The fact that Copland is microkernel-based and offers preemptive multitasking and protected memory space will similarly break down limitations some perceived with previous Mac OS versions. Those interested in designing Macintosh solutions previously available only to high-end design, scientific, simulation, and other workstation customers will be attracted by new Mac OS features like preemptive multitasking, a predictable I/O model (that is, one where you can be certain how long I/O tasks will take to execute), and memory-mapped file I/O. Similarly, the new architecture, particularly interface changes that you can read about in the section "A Customizable Human Interface" on page 8, will encourage development for customers at the low end, who may not know (and don't want to know) the difference between a file system and an operating system.

Making Users More Efficient

Apple engineers have one main strategic end in mind as they reengineer the Mac OS: to let personal computer users be more efficient on the Macintosh computer than on any other platform. In order to do so, they're striving to accomplish two goals:

- to provide a robust, sustainable architecture, one that serves your needs and your customers' needs for the foreseeable future without needing significant revision
- to make the Mac OS closely match the way people want to work; that is, to improve the user experience

To give you a top-level overview of just what Copland will look like when it's complete, the next sections discuss the specific steps Apple will undertake to meet those goals. (If you want a more in-depth look at Copland, see the box "Copland Resources" on page 10 to find out where to go.)

Stable Foundation for the Future

Shipping Copland will be the first part of a two-phase strategy aimed at giving the Macintosh platform a new, robust operating system foundation, one that serves well into the next millennium. The second, and final, step will be the subsequent version of the Mac OS, currently named *Gershwin,* which Apple engineers are developing concurrently with Copland for release later in the decade.

Apple strategists have known for some time that the current version of the Mac OS rests on a foundation that essentially hasn't changed since the days the Macintosh shipped with 128K (yes, that's kilobytes) of memory. The Macintosh community's needs have changed a bit since the original Macintosh foundation was laid in the 1980s; users have grown in number and sophistication, and, at the same time, Apple has adopted a variety of hardware innovations—most obviously the PowerPC microprocessor and—later this year—the Peripheral Component Interconnect (PCI) hardware expansion standard.

System 7, and older Mac OS versions, have served well throughout all the change. When you think about it, it's pretty impressive that the same application you wrote for an 8-MHz 68000 Macintosh computer still works pretty much the same way on today's 110-MHz PowerPC 601 speed demons. The current situation leaves Apple and the Macintosh development community with, on the one hand, dramatically more sophisticated customers and hardware technology. On the other, hand, though, we have an OS foundation that makes it an increasingly difficult challenge to meet more complex customer needs and take advantage of the hardware's inherent advantages.

Preemptive Multitasking for Free

The heart of the matter is that the current Mac OS is terrific at managing the hardware through simple, singular tasks. It could be better, though, at getting the hardware to perform multiple tasks at the same time. The current Macintosh architecture is too synchronous to allow for optimal processor sharing; although there are some exceptions, the computer can't go from one task to another until the first task is complete. (Actually, some preemptive multitasking can be made to happen under the current Mac OS through VBLs—or vertical blanking interrupts—but they're difficult to use and are useful only for small, repetitive tasks executing out of main event loops that neither allocate nor release memory.)

Copland is being engineered to better use system resources through concurrency. With a new I/O model and file system, a different way of managing interrupts, a more tightly integrated networking and communication subsystem, multiple address spaces, and a preemptive microkernel, the new operating system architecture can provide concurrency and make effective use of increasingly more robust microprocessors and systems.

Essentially, I'm talking about preemptive multitasking here. The current Mac OS provides for *cooperative* multitasking, which takes place when software makes a WaitNextEvent call and surrenders its control to another task. However, if software doesn't use WaitNextEvent, it can theoretically occupy the processor forever, never letting any other tasks have processor time.

Copland will, for the first time, provide extensive *preemptive* multitasking services for Macintosh software, where one task doesn't have to depend on another task's "good will" to get processor time. The Copland architecture provides a protected memory space in which preemptive tasks can run; while preemptive tasks execute, the file system, networking, and device I/O will remain available, letting users do more than one thing at a time.

The Toolbox will continue to run your applications in a cooperative multitasking environment. (See the illustration "Cooperative Toolbox, preemptive tasks" on page 7.) With Gershwin—the next major Mac OS release—the Toolbox itself will become fully preemptive, running each application in its own memory space.

"Why can't I have complete preemptive multitasking for my application right now, with Copland?" you might be asking. Because it would break most existing software, which would be a bad thing for all of us. Instead, Apple is giving you time to get used to the idea of preemptive multitasking and an operating system with multiple address spaces. For Copland, you can isolate certain parts of your software's code as "tasks" that can execute preemptively in the protected memory space, for example, moving data from a scanner to a hard disk, doing network file transfer, or indexing. By adopting Copland's features now, your product will get full preemptive multitasking "for free" when Gershwin's preemptive Toolbox environment is made available later in the decade.

New Run-Time Environment

You might recall my saying earlier in this article (some of you with disbelief verging on horror!) that Copland will break existing system extensions and DAs. To get philosophical for a moment, there is a reason for everything. In this case, the reason is even a good one: a new run-time environment that eliminates a perceivable weakness in the original Mac OS architecture and that "should dramatically improve the stability of the OS," to borrow the words of *MacWeek*'s Robert Hess from the April 10, 1995, issue.

Under the current run-time environment, system extensions are loaded into the system at startup time. Once loaded, they patch various Toolbox routines to carry out their tasks. In doing so, extensions can alter the behavior of the Mac OS and applications, sometimes in ways that get in the way of other tasks, or even create hangs and crashes.

Copland is based on dynamic linked libraries (DLLs), and it has no system-wide trap dispatch table. Additionally, with Copland, system extensions simply won't be loaded at startup. Current extensions won't work, and any application's features that rely on those extensions won't work, either.

What can you do to get your product to work in the new environment? First off, be sure your software doesn't make any use of system-wide patching. Those of you who have been around for awhile know that patches are an exception to the original Macintosh programming rules; so many people started to use them, though, that they became an official, documented way of carrying out certain tasks. By eliminating the current patching mechanism and replacing it with one that will better manage patches, Apple is providing a more stable, clean environment that will run your applications more predictably. This may take some work on your part, but your customers will experience a much-improved system as a result.

If you're a developer of system extensions, there are several things you can do. Details about the activation model for extensions under Copland will be made available at the 1995 Worldwide Developers Conference (WWDC). Videotapes and audiotapes of WWDC sessions will be available to you after the conference; to find out how to order them, see "Copland Resources "on page 10. *[Editor's Note: This issue of* Apple Directions *went to press two weeks before the developer's conference began.]*

In the meantime, you'll want to clearly separate the activation code from the code that controls your extension's features. When Apple tells you more about the extension activation model, you'll then be able to rewrite the activation code. You might also consider rewriting parts of your extension (or the entire extension, if appropriate) as so-called faceless background applications—applications of type 'appe' that don't use any user interface elements and simply run as background tasks.

If your application relies heavily on an extension, you'll want to eliminate that dependency, or at least isolate from the rest of the application the code for features that depend on the extension. It's likely you'll have to rework the way your application interacts with the extension. (Again, we'll tell you how when we know.) In any case, be sure your application can launch without the extension being present. That way, if you can't completely revise your product by the time Copland is released, your customers will be able to use those features that don't depend on extensions.

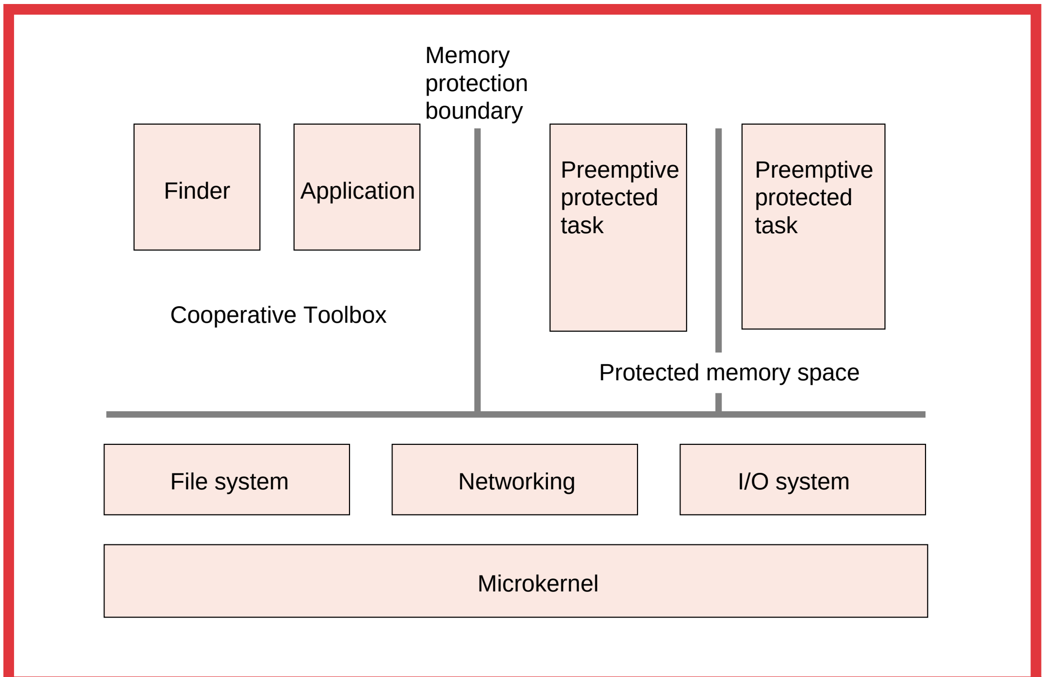
Other Foundation Features

Preemptive multitasking and a new run-time environment are two of Copland's features that will move the Mac OS toward its robust foundation. Here are the additional features that will become part of the Macintosh system foundation for years to come:

- *Native PowerPC processor–based system software code.* Copland will be the first native version of the Mac OS; more than 90 percent of Copland will consist of native RISC code.

Apple had to take a pragmatic approach to the first Mac OS versions that shipped with Power Macintosh computers. In another case of software development lagging behind hardware, Apple engineers couldn't get out a completely native Mac OS in time to ship with the first PowerPC processor–based computers. Instead, they had to spend what time they had taking only certain routines native—the ones that would result in the most noticeable performance increase. By the time Copland ships, Apple's software engineers will have had the time they need to rewrite and optimize the Mac OS to take full advantage of PowerPC RISC performance.

Copland will still run all 680x0 applications under emulation, but the first version of Copland will



Cooperative Toolbox, preemptive tasks. Under Copland, the Finder and applications run under the cooperative Macintosh Toolbox environment, along with QuickDraw GX, user experience features, and the Toolbox itself. Applications will be able to spawn tasks that run preemptively in the protected memory space.

run only on Power Macintosh systems. Apple is still determining its plans to ship a 680x0-based version of Copland.

• *Hardware abstraction layer.* Another difference between the Macintosh systems of yesterday and today is that, until now, the Mac OS ran only on hardware designed and manufactured by Apple. Now Apple has staked its future to licensing, and the Mac OS will have to run on other manufacturers' hardware. To make it easier for the Mac OS to work with Macintosh-compatible hardware—and because PowerPC processor–based hardware architecture requires it—Copland will include a hardware abstraction layer to more distinctly separate the Mac OS from the hardware it runs on. This will also make it easier to be sure your products run on forthcoming Mac–compatible computers, helping to guarantee that you can reach every corner of the expanding market for Macintosh products.

• *New file system, I/O system, and virtual memory model.* Apple's engineers have been contemplating a new Macintosh file system for some time. Under Copland, the file system will be rewritten in native PowerPC code; it will also enable concurrent task execution, that is, letting multiple tasks execute at the same time. Thankfully, it's being written in a way that won't require you to make alterations to your software; existing applications will be able to make use of the new file system. Additionally, it's expected that the new file system will provide a simpler and faster application programming interface (API).

The Copland I/O model will also enable native RISC performance for all I/O operations and concurrent task execution; for example, under the old I/O model, when copying a file the system first reads the file and only writes the file to the new location after completing the read operation. Under Copland, the system will be able to read and write at the same time, speeding such operations by at least 50 percent.

Existing drivers, however, won't work under the new model, requiring all drivers to be rewritten if they're to work with Copland-based computers. If you write drivers and you want to get a jump on preparing your products to work under Copland, see page 9 under "Here's What You Can Do."

Additionally, Copland employs a new model for virtual memory management designed to reduce customers' RAM requirements by enabling software to make much more efficient use of existing RAM. For details about the file system, I/O subsystem, and virtual memory management, read Tim Dierks's article, "Copland: The Mac OS Moves Into the Future," in *develop,* Issue 22. (You can find *develop* on the June Reference Library Edition of the Developer CD Series.)

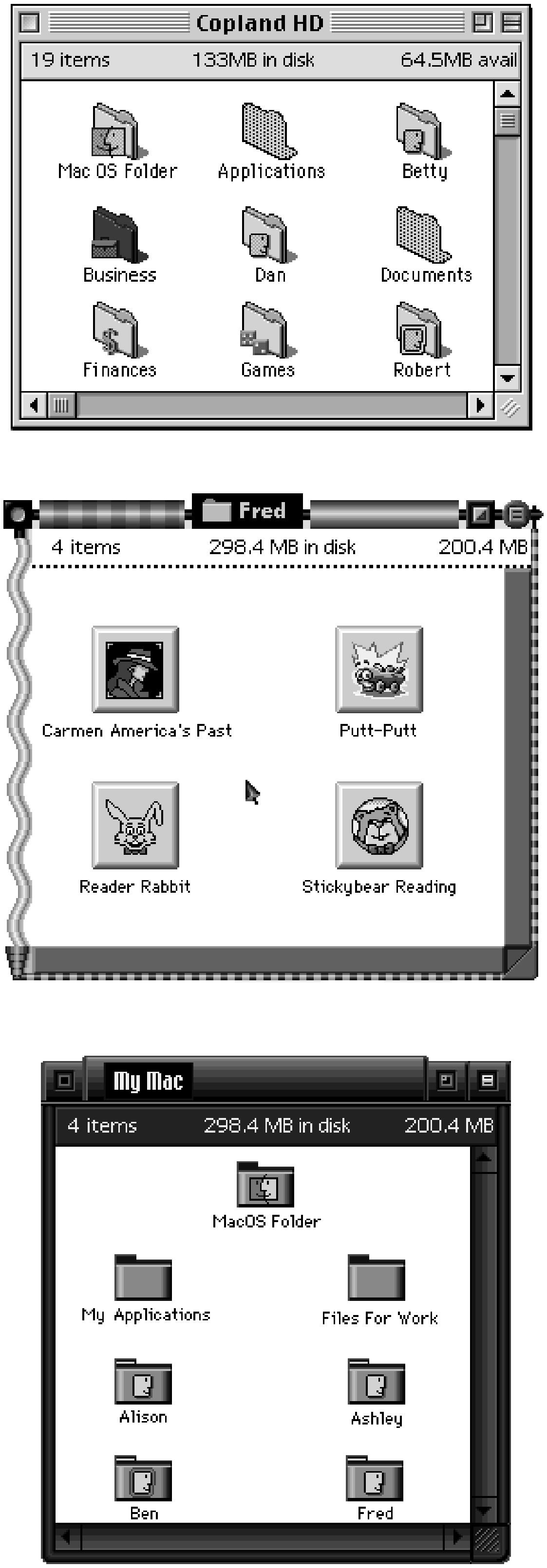
• *OpenDoc support.* OpenDoc will have been available for some time by the Copland ship date next year. Copland will fully support the new component software architecture, introducing a new application run-time environment for OpenDoc parts that will comply with the Code Fragment Manager and the System Object Model on which OpenDoc is based.

• *New networking model.* Open Transport is Apple's new, fully native networking and communication architecture. It consists of new implementations of every aspect of Macintosh networking and communication, including new APIs, a new way of working with the underlying operating system, and new human interface features.

You'll be able to write one networking application to the Open Transport APIs that will then work with multiple networking protocol stacks. All existing Macintosh networking applications will continue to work with the new scheme, although they won't have access to the new Open Transport features, which include native Power Macintosh performance.

With Copland, Open Transport will be a part of the I/O architecture, so that networking tasks can be completed simultaneously with other tasks. For example, even while the computer copies files from a file server on the network, its processor will remain available to complete other tasks. Under System 7, users would have to wait for both reading from the file server and writing to their own hard disks before being able to do anything else on their systems. Apple just shipped the beta version of Open Transport; for details, see the news story on page 12.

• *Unicode and international support.* Copland will offer significant new support for internationalized software, offering developers three different approaches to



Copland human interface styles. The style on top (showing the disk Copland HD) is Copland’s default style.

shipping “world-ready” applications. WorldScript, first offered with System 7.1 and steadily enhanced since then, will be native and completely integrated with the underlying OS (without relying on patching). Also, Copland will offer new ways to represent international text, including true Unicode support in major system components and a new object-based text model that combines the best features of WorldScript and Unicode.

Like all versions of the Mac OS since System 7.1, Copland will support all modern writing systems, including writing systems for languages that are read from top to bottom or right to left, and complex contextual writing systems such as Hindi. In addition, Copland will enable customers with multiple language needs to include any combination of writing systems within a single document or application. Copland will also provide enhanced support for market-specific features, including advanced support for Asian input methods, user-defined character support, and an open font architecture, enabling multiple font formats to coexist.

Creating the Best User Experience

The second major goal of the Copland release will be to create a new, radically better Macintosh user experience, making it more closely reflect the way people want to work with their computers. Copland designers are doing their part to assure that the Macintosh user experience is simply the best personal computers have to offer.

Apple has spent years talking to customers and finding out what innovations would make the Macintosh even easier (and more fun!) to use. With Copland, the Mac OS interface will be significantly updated, one that will again distinguish it very favorably from any other graphical user interface. Copland will also introduce a variety of tools and interface elements that Apple can’t wait to spring on personal computer users.

A Customizable Human Interface

Perhaps the most noticeable innovations—certainly the most fun ones—are that computer users can control the way they work with the system, both the way it looks (and sounds), and the levels of features they work with. In designing Copland, Apple engineers faced a dual challenge: On the one hand, they wanted to be sure the Mac OS continues to appeal to an installed base of customers that’s growing increasingly expert at working with computers. On the other, they needed to ensure that the Mac OS would appeal to customers in today’s fastest-growing market segment—home users, many of whom don’t know much about how computers work, and don’t want to know much more. The engineers’ answer: a scalable, customizable Finder.

First, Copland designers had to be sure customers could determine the level of usage that would be appropriate for them based on their ability, experience and interest. As a result, Copland customers will be able to choose the extent to which they use Finder commands. Beginning users can decide to give themselves access only to basic commands, keeping their computing experience as simple as possible. On the other end of the spectrum, power users can enable all system commands and avail themselves of all the power that comes with Copland’s advanced features. In between, intermediate users can select the user level appropriate to them.

Further, Copland designers wanted to satisfy customer’s increasing demands to be able to change the way the system looks. As a result, Copland customers will be able to choose from a variety of Finder human interface styles. Copland's default user interface style, while reminiscent of System 7, will be noticeably different, employing mock three-dimensional shapes and other new elements, as well. The default style, and two of the optional styles, are shown in the illustration "Copland Human Interface styles" on this page. Users will be able to further adapt these styles to better reflect the way they want to work.

New Navigation, Easier Access to Information

Almost as noticeable will be new navigation tools and other design modifications that will give users easier access to information. First off, and perhaps most importantly, the Copland Finder is completely threaded; all Finder processes will now run under cooperative multitasking, making them noticeably faster.

Further, Copland gives users a way of working with data (files, folders, applications, and so on) that bypasses the file system's hierarchical nature. For this purpose, Copland will introduce a new human interface element currently called a *viewer.* Viewers are folder-like containers that will let users store their folders and files by attribute (such as name, date created, kind, and so on); users select the attributes that determine what items are placed in the viewer. Setting up viewers works a little like the way system searches work today, but instead of returning a list of the location of files and folders that match the search criteria, the search sets up a viewer containing the files and folders. Each viewer automatically updates itself to include newly created files and folders that match the viewer's criteria.

Another useful new feature will make copying much more convenient; when users drag a file over a folder to copy it to the new location, if they hold the mouse button down when the file touches the folder, the folder opens and they can place the file in any other folder nested within the first folder. Also, if they are unhappy with the results of the operation, they can simply perform an Undo command (by typing Command-Z), and the file returns to its original location.

Then, of course, there's the new, much more robust and easy-to-use standard file dialog box used in Open and Save operations. You'll want to take a look at the illustration of it below.

Active Assistance

Apple Fellow and User Experience Architect Don Norman formally introduced the concept of active assistance at last year's Worldwide Developers Conference, announcing that the Macintosh interface would increasingly attempt to anticipate users needs in forthcoming versions of the Mac OS. Apple Guide, Apple's advanced online assistance architecture introduced with System 7.5, implemented the first part of that vision.

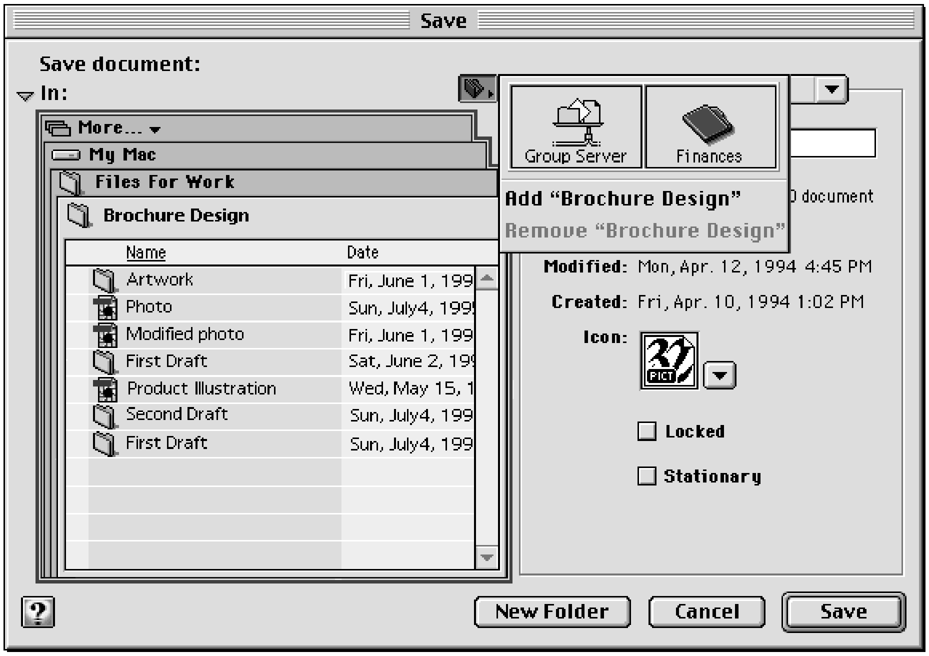
Copland carries the vision several steps further with agent-based interface elements that automate routine tasks as well as the advanced "do it for me" Apple Guide feature. With Copland's "do it for me" active assistance, users can automate virtually any routine system task—such as setting up a shared hard disk on a network, organizing e-mail, or compiling and distributing a weekly status report.

As an example of agent-based automation, say a small business owner wants to automate the task of preparing reminder notices for overdue customers. The owner schedules a Copland agent to automatically launch an invoicing system, find customers whose payments are overdue, and mail-merge their records with a reminder notice. Every Friday, the assistant generates and prints reminder notices automatically.

For more information about Copland's active-assistance interface and its other user experience enhancements, you can read Apple's official Copland overview, "An Introduction to Copland—The Mac OS Foundation for the Next Generation of Personal Computers"; see the "Copland Resources" box on page 10 for its location on AppleLink and the World Wide Web.

Here's What You Can Do

That, in a rather large nutshell, is what Apple hopes to accomplish with the release of Copland. If you're like most people who hear about it, you'll want to get started right away to get your products ready for Copland. Apple is still determining its developer release plans, and we'll let you know specifics as soon as we learn them, ourselves. Fortunately, though, there are a number of important steps you can take in advance of the first Copland seed. I'll list all of them here, even though some have been mentioned already.



Save document:

In:

Name	Date
Artwork	Fri, June 1, 199
Photo	Sun, July4, 199
Modified photo	Fri, June 1, 199
First Draft	Sat, June 2, 19
Product Illustration	Wed, May 15, 1
Second Draft	Sun, July4, 199
First Draft	Sun, July4, 199

Modified: Mon, Apr. 12, 1994 4:45 PM

Created: Fri, Apr. 10, 1994 1:02 PM

Icon: PICT

☐ Locked

☐ Stationary

The Copland standard file dialog box will let users open and save files much more efficiently than before.

First, be sure you read, look at, and listen to everything you can get your hands on about Copland. The more you know, the more effectively you'll be able to plan for the new Mac OS. For a listing of available information about Copland, see the box "Copland Resources" below.

To begin ensuring that your product will work under Copland, you'll want to do the following:

- Go native! It will be more important than ever that you port your application to run in native Power Macintosh mode once Copland is released. Native software will be able to take full advantage of Copland's performance enhancements, and many of its other features, as well.
- Eliminate any code from your software that relies on undocumented data structures; many undocumented, low-level behaviors will no longer be supported under Copland, and if your code uses them, it will break.
- Isolate file-management code into concise modules so you can be ready to take advantage of the new file system.
- Plan on eliminating system-wide patching from your code; you can start by avoiding patching where possible. I know that your current software has to use system-wide patching, but that won't be supported under Copland. If you want your application to work under future versions of the Mac OS, at some point in the not-too-distant future you'll have to make the tough call to rewrite it without system-wide patching (although local patching will continue to be supported).
- Make sure your application's dependency on any extension is "soft," so the application can launch even if the extension isn't present. This will give Copland customers at least partial access to your application's features in case you can't have a fully Copland-compliant version of your product ready by the Copland ship date.
- If you develop system extensions, separate the actual functionality of an extension from the way it's activated. There'll be a session on Copland extensions at this year's WWDC, and you can order a recording of it. (See "Copland Resources" to find out how.) Once you receive details about the Copland extensions activation model, you can then write the activation code—without having to alter the features code, we

Copland Resources

The following resources are currently available to help you understand and get ready for Copland.

- "Copland: The Mac OS Moves Into the Future," in *develop,* Issue 22. This article, written by the former technical lead for Copland, Tim Dierks, provides technical details about the Copland architecture. Issue 22 of *develop* can be found on the June Reference Library Edition of the Developer CD in the Periodicals folder.
- *An Introduction to Copland—The Mac OS Foundation for the Next Generation of Personal Computers.* This brochure offers a high-level look at Copland's features and describes Apple's business reasons for introducing the next-generation Mac OS. You can find an electronic copy of it on the Internet's World Wide Web at the location http://www.info.apple.com/macos/rels/releaseover.html/copland/; it's also available on AppleLink in the Apple Directions area (path—Developer Support:Developer Services:Periodicals:Apple Directions:Apple Directions June 1995).
- Tapes from 1995 Worldwide Developers Conference (WWDC) Copland sessions. Apple will present the following sessions about Copland at this year's WWDC:

Copland: Essentials of the Next Mac OS. Describes the design philosophy of Copland and the business opportunity for Apple and developers. Defines the key components of Copland, how they fit together, and the role of Copland in the context of Apple's current strategies for System 7.5 and OpenDoc.

Copland's Application Model. Provides an overview of the architectural differences between System 7 and Copland. Gives insight into the OS feature set and how your products can leverage Copland's capabilities to become simpler and more robust while delivering greater performance.

Information Access in Copland. Describes Copland's mechanisms for improving information presentation and access. Gives details about Copland's file system.

Copland's Integral Support for the World. Describes how to localize applications for all markets and enhance Mac OS multilingual capabilities under Copland. Gives details about Copland's new text model, which supports Unicode, and of the new mechanisms for language switching, sorting, date/time support, and text services.

Copland's Customizable User Experience. Describes how Copland users can adapt the user experience to meet their individual needs. Gives details about Copland's switchable appearances, new Toolbox capabilities, interface scalability, and support for multiple user configurations.

Copland: Toolbox Customization. Describes how you can customize the Copland Toolbox to provide unique user interface elements. Shows how to create Interface Definition Objects (IDOs) and drop-in panels. The Toolbox's use of System Object Module (SOM) is also described.

Copland's Active Assistance. Provides an overview of Copland is active assistance architecture, as well as details about assistants, automating repetitive tasks, and making your software assistance-aware.

Copland: Evolution of System Extensions. Describes the new Copland run-time environment and how extensions will fit into it. Details about Copland patching, the replacement for the INIT loader, and new facilities for managing system extensions for individuals or multiple users will be presented.

Copland's Modular I/O System. Details Copland's I/O system. Describes how to create drivers, and how various services provided by the system will support I/O components.

Planning for Copland. Presents strategies for product development to help ensure a smooth migration to Copland. Emphasizes techniques that will improve compatibility and minimize the effort to take advantage of Copland advances.

Information for ordering WWDC session tapes will be available on AppleLink after the conference ends on May 12 (path—Developer Support:Developer Services:Events/Marcom:WWDC).

- *Inside Macintosh* overview chapters. Overview chapters from forthcoming volumes of *Inside Macintosh* that document Copland were made available at the 1995 WWDC. If you didn't get the WWDC CD, you'll be able to find these chapters (and other Copland documentation, as it's released) on forthcoming editions of the Developer CD.

hope. Also, if appropriate, rewrite system extensions (INITs) as background faceless applications (applications of type 'appe').

- Rewrite desk accessories as small applications. Current DAs are not supported under Copland, so you'll need to substantially revamp them to work with the next-generation Mac OS. (Hey gang—we've only been telling you to do this since the introduction of System 7. It's finally time to bite the bullet and do it!)
- Study the PCI model for writing drivers. As I said before, the Copland I/O subsystem will require existing drivers to be rewritten, although Apple intends for all PCI drivers to work unmodified under Copland. Understanding the PCI model will at least set your thinking in the right direction until Apple publishes complete specifications of the Copland driver model. And if you write a PCI driver, you can be sure it will work with Copland. To get started, see the August 1994 *Apple Directions* article "PCI Cards for the Second Generation of Power Macintosh," which refers to other PCI resources. (That issue, and every other back issue of *Apple Directions,* can be found on the June System Software Edition of the Developer CD as well as on AppleLink [path—Developer Support:Developer Services:Periodicals:Apple Directions].) Also, you can purchase a recording of the 1995 WWDC session that covers the Copland I/O system.
- To get ready for preemptive multitasking, remove dependencies between different parts of your application. Also, thread your application using the Thread Manager. Under Copland, applications can create kernel tasks (or threads) that run preemptively in the protected memory space. Details on exactly how your application can be revised to do this will be released soon; however, if you've fully adopted the Thread Manager, you'll already have thought through the issues you'll need to face to let your application use Copland preemptive multitasking services. The Thread Manager is available as part of the Mac OS SDK (Software Development Kit), which is available from APDA. (See page 32 for APDA ordering information.)
- Adopt AppleScript. Get your application ready to take advantage of Copland active assistance by making it scriptable—that is, Apple event–aware. We've been telling you this under one guise or another for some time now. Given that adopting the Open Scripting Architecture Object Model will also help you turn your software into OpenDoc parts, it's more important than ever that you do so. For more information about AppleScript and related technologies, see "System 7.5: Apple's Unified Operating System for 680x0 and Power Macintosh Computers" in the June 1994 issue of *Apple Directions.*
- To get ready to use the Apple Guide "do it for me" feature, study Apple Guide and adopt its current features in your existing products. Apple Guide is also part of the Mac OS SDK.
- If you're a networking and communications developer, revise your applications according to the Open Transport APIs so they can give your customers access to advanced Open Transport features. A beta version of the Open Transport SDK is available on this month's Developer CD. (See the news story on page 12.) The final Open Transport SDK will be available from APDA by mid-year.
- A few other recent Apple technologies—QuickDraw GX, PowerTalk, and Macintosh Drag and Drop—will also be important parts of Copland. You'll want to adopt them now, if you haven't already; both are included in the Mac OS SDK.

Lay Your Own Foundation for the Future

If all of the above steps sound like a lot of work to you, rest assured that they are! But keep in mind that the customer release of Copland won't take place until some time in the first half of next year. Fortunately for you, this gives you a bit of time to do all the work necessary to adopt Copland technology and be sure your products provide your business a foundation for the future. Remember, though, just how quickly a year can pass. We're telling you about Copland this long in advance of its release to help you start getting ready for it today. We want to give you the information you'll need to beat the competition likely to be drawn to the platform because of its modern capabilities and radical new features. And we want you to be among those introducing their products along with Copland sometime next year. ♣

The author wishes to thank Ken Bereskin, Apple new technologies evangelist, for his generous help preparing this article.

Apple News

They're Heeeeere…

continued from page 1

provide you with a larger market for your Mac OS-compatible products. Apple also plans on gaining market share (and selling its own computers) through its "stand out/fit in" technical innovations, cross-platform compatibility, and aggressive price/performance.

The Power 80, 100, and 110 Computers

The new Power Computing systems, available in desktop and tower configurations, are based on the PowerPC 601 80-MHz, 100-MHz, and 110-MHz microprocessors (see "Power Computer Configuration Details" on page 12). They are comparable to Apple Computer's Power Macintosh 7100 and 8100 class of computers and have been certified as fully Mac OS–compatible by Apple.

Innovations in the Power 80, 100, and 110 computers include

- an optional quadruple-speed CD-ROM drive
- competitive pricing
- easy system upgrades through an expansion board
- both 15-pin Macintosh and VGA video connectors
- extended keyboard included with every system
- a dual-channel, asynchronous SCSI interface, for easy RAID (redundant array of inexpensive disks) implementation
- bundled software with each system: ClarisWorks 3.0, Quicken 5.0, Now Utilities, Now Contact, Now Up-to-Date, and a collection of 300 TrueType and Type 1 fonts from Bitstream
- a 30-day money-back guarantee to customers
- lifetime toll-free technical support
- one-day delivery of "semi-customized" computer configurations, direct from manufacturer
- customer choice of desktop or tower enclosures

Availability and Pricing

The Power 80, 100, and 110 computers will be shipping in limited quantities beginning May 1, 1995, in the United States. Pricing for the desktop systems will range from $1,995 to $2,899, depending on the configuration of the unit. Power Computing's tower systems will begin shipping June 15, 1995, with pricing to be announced at a later date. All systems will be available directly through Power Computing's mail-order operations.

OpenDoc Developer Release 2

Apple Computer, Inc., has released Developer Release (DR) 2 of OpenDoc as part of its June 1995 Developer Information Mailing, which ships with this issue of *Apple Directions.* This is the version of OpenDoc that you should use to develop your OpenDoc parts, and/or OpenDoc-compliant versions of your existing applications in anticipation of the first customer release of OpenDoc in the second half of 1995.

OpenDoc DR 2 contains CA Lib, code that you can use to help turn your existing applications into OpenDoc containers that can accept content from OpenDoc parts. It's the first feature-complete, stable build of OpenDoc; all APIs (application programming interfaces) have been frozen. In addition, this release ships with comprehensive documentation.

The OpenDoc DR 2 CD-ROM also contains information about how you can obtain the Open Parts Framework (OPF), Apple's framework for OpenDoc development. OPF is expected to ship shortly after OpenDoc DR 2.

If you haven't received your copy of the OpenDoc DR 2 CD, you can obtain one by sending an AppleLink message to OPENDOC.

Power Computer Configuration Details

Processor	PowerPC 80: 80 MHz; Power 100: 100 MHz; Power 110: 110 MHz
Cache	256K Level 2 cache (512K and 1 MB also available)
NuBus™ slots	Three
Memory	8 MB, expandable to 200 MB
Video support	Built-in video port, supporting Macintosh and VGA monitors up to 17 inches; optional high performance video card (card is included in Power 100 and 110 configurations)
Floppy disk drive	3.5-inch 1.44 MB self-ejecting floppy disk drive (Apple SuperDrive–compatible)
Hard disk	Internal 3.5-inch drive, 356 MB to 4 GB sizes available
Internal hardware bays	Space for one 5.25-inch full-height drive and two 3.5-inch drives, or for four 3.5-inch drives
Quadruple-speed CD-ROM drive	Optional
Networking	Ethernet, two serial ports
Audio	44.1-kHz, 16-bit sound input and output; micro phone and headphone jacks
I/O ports	Dual-channel SCSI interface; ADB port for keyboard and mouse

Apple Ships Open Transport Beta Software

The beta version of the new Mac OS networking and communication system—Apple Open Transport—shipped on the June 1995 Reference Library edition of the Developer CD.

Open Transport provides a set of application programming interfaces (APIs), a new way of working with the underlying OS, and human interface features. By writing to the Open Transport APIs, you can write a single version of networking/communication software that can then work with multiple protocol stacks and network connections.

Existing Macintosh networking and communication software will work with Open Transport, although it won't have access to its advanced features, which include the following:

- For developers, Open Transport makes it easier and more cost-effective to develop Macintosh-based applications for a wide variety of customers. With Open Transport, the Mac OS has a built-in networking and communication environment that is based on cross-platform industry standards, such as the POSIX-compliant X/Open Transport Interface, UNIX® STREAMS, and Data Link Provider Interface. Applications written to support Open Transport automatically support a wide range of networking environments (including serial, dial-up network, LAN, and WAN), and multiple protocols (AppleTalk, TCP/IP, serial, and others).
- For network managers, Open Transport provides significant management capabilities, such as new flexibility in setting up network configurations; with Open Transport, the network manager can determine configuration settings for users on the network or allow the users to determine their own settings. Open Transport also supports the Dynamic Host Configuration Protocol (DHCP), a new standard that allows network managers to administer addressing and other configuration information for TCP/IP networks from a central location.
- For individual customers, Open Transport provides two important benefits. First, it simplifies switching from one network configuration to another. For example, a notebook computer user might hook up to the Internet in various locations, each requiring a different network configuration. With Open Transport, settings for each network location can be stored in a file for easy access and use. Second, Open Transport integrates online help and active assistance, based on Apple Guide technology, to make it easy for the individual to hook up to a network without the assistance of a network manager.

The Open Transport software is written in "native" PowerPC code, and it provides new levels of performance for applications that are accelerated for Power Macintosh. Users should see significant improvements in networking speed for transferring files and in using high-performance applications and high-performance datalinks such as 100 MB Ethernet and ATM (Asynchronous Transfer Mode).

Several hundred developers are actively working with Apple on Open Transport development efforts. Among the developers already endorsing Open Transport, and the products that will employ the new technology, are the following: The AG Group (Net Watchman); Dantz Development Corporation (Retrospect); Farallon Computing, Inc. (Timbuktu); InterCon Systems Corporation (TCP/Connect II); Metrowerks Inc. (CodeWarrior and PowerPlant); Northwestern University (NewsWatcher); Novell, Inc. (NetWare protocols and client for Mac OS); AGE Logic, Inc. (PacerShare/PacerPrint, PacerTerm); SoftArc Inc. (FirstClass); StarNine Technologies Inc.; University of Michigan (X.500); Vicom Technology (VICOM RunTime, MultiTerm, Pro SDK); The Wollongong Group (PathWay Access); Walker Richer and Quinn, Inc. (Reflection for Macintosh); and Wall Data, Incorporated (SNA•ps).

Apple plans to ship the final version of Open Transport to customers in the second half of 1995. More information about Open Transport, including technical documentation, is available by anonymous ftp on the Internet at ftp://seeding.apple.com/public/opentransport/ot_docs.

Apple Reduces Prices on Developer University Self-Paced Training Products

Developer University (DU) self-paced training products just got 30 percent cheaper. DU has also added a new self-paced tutorial, *Intermediate Programming: 7.5 Topics*—a workbook-based tutorial that assists programmers in extending their applications to include the new capabilities of Macintosh System 7.5.

DU also offers more than 20 classroom courses covering a complete range of development topics, including Newton, multimedia, and OpenDoc. DU provides this training at its facilities in Cupertino, California, and selected field-training locations, on an on-site basis, and through international training events.

DU self-paced courses (and their new prices) include the following:

- *Macintosh Programming Fundamentals.* Price reduced from U.S. $595 to U.S. $395.
- *Intermediate Macintosh Application Programming.* Price reduced from U.S. $495 to U.S. $345.
- *Getting Started Bundle.* Price reduced from U.S. $799 to U.S. $559.
- *Object-Oriented Fundamentals.* Price reduced from U.S. $350 to U.S. $245.
- *AppleTalk for Programmers.* Price reduced from U.S. $295 to U.S. $195.
- *Apple Events/AppleScript Programming Tutorial.* Price reduced from U.S. $150 to U.S. $95.
- *Programmer's Introduction to RISC and PowerPC.* Price reduced from U.S. $150 to U.S. $95.
- *Programming With MPW.* Price reduced from U.S. $120 to U.S. $85.
- *Writing and Using Device Drivers.* Price reduced from U.S. $150 to U.S. $95.
- *Intermediate Programming: 7.5 Topics.* This is a new course; its price is U.S. $70.

All Developer University self-paced products are currently available through APDA. For APDA ordering information, see page 32. ♣

Technology

Inside This Section

develop Issue 22: More Than Ever!

Issue 22 of *develop,* Apple's award-winning technical journal, has more content than any previous issue of *develop* and more articles on brand-new—and even future—technology. We're pleased to present our first articles on QuickDraw 3D and Copland, an OpenDoc discussion that provides important background, and much more.

- "QuickDraw 3D: A New Dimension for Macintosh Graphics" introduces Apple's new 3D graphics package for the Power Macintosh, showing how applications can support 3D data.
- "Copland: The Mac OS Moves Into the Future" previews the future of the Mac OS, detailing some of its major components and making suggestions about how to get ready for it now.
- "The OpenDoc User Experience" gives an overview of OpenDoc from the user's perspective—a prerequisite to designing good part editors.
- "Creating PCI Device Drivers" describes the new driver model on PCI-based Macintosh computers, with advice on porting existing drivers.
- "Futures: Don't Wait Forever" provides an update on futures, an invaluable abstraction for applications that handle multiple

please turn to page 22

CD Highlights

Reference Library Edition, June 1995

The lawyers haven't quite finished their dance, so we are not yet able to include Adobe™ Acrobat Exchange LE (the version with the cross-document search engine) on the Developer CD. Also, not all of our technical documentation has been converted into Acrobat format. So, on this disk you'll find documents in both Acrobat and DocViewer formats; please excuse the sawdust and scaffolding as we make this transition, and we'll try to have a consistent browsing and searching user experience for you on the September 1995 Reference Library CD.

Once again, thanks to the large volume of new material on this disc, we had to temporarily remove a few volumes of *Inside Macintosh;* this month, its AOCE's turn. These volumes can be found on the March 1994 Reference Library and New Inside Macintosh CDs, and will reappear on the September 1995 Developer CD. If your subscription begins from June to August 1995 and you need the AOCE books in electronic form but do not have access to the NIM CD, send send e-mail to AppleLink address DEV.CD or Internet address dev.cd@applelink.apple.com, and we'll send you a copy of the March 1994 CD.

So what, you may be asking, is this large volume of new material on this month's CD? Well, since your jealous co-workers have probably already "borrowed" your WWDC CD, we've included beta versions of Open Transport and QuickDraw 3D for you to play with. For fans of Macintosh Programmer's Toolbox Assistant (MPTA), we've included the QuickView Developer Edition, which will let you publish your very own skanky APIs. And much, much more. . . .

ABS Technical Notes

ABS Technical Notes contain the latest bugs, tips, and tricks for developers of software for Apple Workgroup Servers, AppleSearch, AppleShare, AppleTalk Remote Access, and IP Gateway.

APDA Tools Catalog

APDA is Apple's worldwide source for hundreds of development tools, technical resources, training products, and information for anyone interested in developing applications for Apple computer platforms. Customers receive the *APDA Tools Catalog,* which features all current versions of Apple development tools, as well as the most popular tools from third parties. APDA offers convenient payment and shipping options, including site licensing.

Developer Notes

Included here, along with our regular archive, are developer notes for three new products: the Power Macintosh 5200/75 and 6200/75 computers, and the LaserWriter 4/600 PS printer.

Developer Univ. Course Info

This package provides information on the types, cost, and location of training available to Apple's developers through Apple's Developer University. It enables you to determine which training will be appropriate to your needs and how to obtain it. It includes current course descriptions, class dates and locations, and information about obtaining self-paced materials.

Disk I/O Performance Tools

This folder contains SCSI Monitor and IOTracer, performance tools designed to record and report information on file system and SCSI performance in a Macintosh environment.

please turn to page 21

OpenDoc—One Architecture Fits All

Provides Best Solution for Mac OS, Windows Compatibles

Are you thinking about the coming trend in component software?

Does the word *OLE* mean more to you than a bullfighter's call? (And does the prospect of entering into that arena fill you with equal parts of dread and resignation?)

Would you like to save yourself a lot of grief and make better products in the process?

If you answered "yes" to any of these questions, keep reading.

As you may know, OpenDoc will be available on the Mac OS–compatible platform this fall, and the Windows-compatible version will come soon thereafter. You may also know that the two main component-software architectures vying for your attention are OpenDoc (from Component Integration Laboratories, or CI Labs, founded by Apple, IBM, and Novell) and OLE, or Object Linking and Embedding (from Microsoft).

Both OpenDoc and OLE are (or will soon be) available on both the Mac OS–compatible and Windows-compatible platforms. So right now you may be thinking, "Which of these do I commit to? OLE is proprietary and a nightmare to program, but Microsoft has the largest market share. OpenDoc is not controlled by Microsoft and is more versatile, but will it become the dominant component-software model? Should I support both or only one? If I support only one, am I looking at a real bullfight on my hands?"

Relax. You can use OpenDoc to build component software for both platforms, and here's what you'll get for your efforts:

- a development process that allows you to create software that's compliant with OLE 2.0–without having to learn the complexities of OLE
- sales to both markets with only one development effort (well, with a *bit* of extra work—depending on your approach and situation, you may have to write platform-specific code that draws the content of your windows and manipulates menus)
- an architecture that is more powerful than OLE 2.0 (making it possible for you to write better software)

Just to make sure I say it plainly: Using OpenDoc, you will be able to use the same source code to create OpenDoc parts that will run on both Mac OS– and Windows-compatible computers. Furthermore, on both platforms, *OpenDoc parts will work as well as or better than OLE objects inside any document that accepts OLE objects, and customers can embed OLE objects in OpenDoc documents with equal success.*

(And, by the way, unless I specifically say "OLE 1.0," every reference to OLE in this article is to OLE 2.0.)

What's the Catch?

In a word, there isn't one. Or if there is, it's on Microsoft's side. Here's some background information you should know. This OpenDoc/OLE technology is working today. (In fact, if you were at the 1995 Worldwide Developers Conference, you saw it in action.) Novell has developed this technology, which they call *ComponentGlue,* and has it working quite well on Windows-compatible computers. Actually, they've been demonstrating it for almost a year—see the screen shot on page 6 of the November 1994 issue of *Apple Directions.*

(As this article went to press, Apple was porting Novell's source code to provide ComponentGlue for Mac OS–compatible computers. ComponentGlue, whose source code Novell will donate to CI Labs, is an optional "interoperability service" of OpenDoc that you can leave out if your software does not have to interact with OLE.)

If you port your OpenDoc part to the Windows platform, how much extra work is it to ensure that your part will work in OLE documents (which, in OLE terminology, are called *containers)?* Zero. None. *Nada.*

Well, almost *nada.* With absolutely no extra work, your part will be visible to OLE applications, but it will have a generic name—something like "OpenDoc Part." By adding one line of code to a resource file to register the part with OLE, the part will appear to OLE applications with its own name ("GerbilCalc Spreadsheet," for example) and will be indistinguishable from any other OLE object type.

"But," you say, "I'll bet an OpenDoc part doesn't work as well inside an OLE container as it does inside OpenDoc." Actually, it does. Once an OpenDoc part has been activated, there's zero overhead—the mouse and keyboard input are "wired" directly to the part-handler code, bypassing OLE completely (more on this later).

The Catch

So where's the catch I mentioned? If it's anywhere, it's in the fact that, in some few situations, OpenDoc parts don't behave exactly the same in OLE containers as they do in OpenDoc documents. But the problem lies in OLE, not OpenDoc.

The capabilities of an OLE object are influenced by the capabilities of the OLE container around it. Because of this, an OLE container sometimes prevents an OpenDoc object from exhibiting behavior that it would inside an OpenDoc document.

For example, an OpenDoc part is "live" even when it is not the part that the user is currently working with (that is, it doesn't receive keyboard and mouse input).

Because of this behavior, a part that mimics an analog clock has a second hand that moves each second, even when the user is, say, typing text into a text part. In most of today's OLE container applications, only the currently active object can change—all the other objects are represented by "frozen" images of the last time they were active. So if you put an OpenDoc clock part in an OLE container, the second hand won't move once a second—but not through any fault of ComponentGlue.

(FYI, an OLE object must be active for it to be dragged. In contrast, OpenDoc parts are draggable at any time.)

OLE Servers

Before I can explain how OpenDoc works on Windows-compatible computers running OLE, I have to explain a bit about how software runs under OLE. In OpenDoc, a part handler is the code that executes when the user clicks on a part of a given type. Click on a text part, for example, and the text part handler executes. This part handler is responsible for responding to keystrokes and mouse clicks that are related to that part.

please turn to page 18

Human Interface

Network Nirvana

By Peter Bickford

As I was sifting through my correspondence, looking for an interesting letter to run at the start of this month's column, one in particular caught my attention. Unfortunately, it seems the writer was more than a little steamed when he wrote it, and the resultant heat of his "flame" message caused the text to be partially incinerated before I could copy it down. The following is my best reconstruction from the cinders that remained:

Dearest Doktor,

I enjoy reading your column every month, finding it a source of endless enlightenment. If it's not too much bother, I wonder if you could aid me in an interface problem that has me—well, I must admit—a bit vexed.

I have been using AppleLink for years and have found it a reliable and easy-to-use system. Still, it has been some time since AppleLink was released, and the state of the interface art has advanced greatly in the intervening years. In this bright new age of human interface, I had hoped that eWorld, America Online, or one of the many other systems that have appeared would lead me to network interface nirvana. Alas, I fear my hopes have been shattered. Can you help?

—A Beleaguered Soul

As a side note, I was amazed at how much easier it was to reconstruct the letter having known something about the sort of people who write me with interface questions. The Human Interface column universally attracts intelligent individuals with the most refined sensibilities; thus, the fragments of letter that appeared to say things like "Couldn't those jerks get someone with a clue to come up with an actual interface for these systems . . . ?" were obviously not what they appeared to be at first glance. Although the deathless prose of the original is lost to us, I believe I have been able to salvage at least the spirit of what was being said. I might suggest to this gentle reader in the future, however, that he turn down the heat a bit before clicking the Send button.

Despite all this, the reader raises some very good points. The state of the art in online systems leaves an awful lot to be desired—ranging as it does from "clunky" to "downright unusable." Over ten years have gone by since AppleLink first came out. During that time we've seen graphical user interfaces become nearly universal, and the average modem speed increase by an order of magnitude or more. So where is the next great generation of online user interfaces?

What was acceptable ten years ago is not acceptable now. It's time for us to raise our standards and design the next generation of online user interfaces. Here's a few ideas to get things started.

Get the User Out of the Network Configuration Business

Users generally aren't interested in using the network—they're interested in using information or services that happen to be on the network. They have no interest in frame types, parity, handshaking, or protocols. They just want the whole thing to work.

We can start to do the right thing by sparing our users the murderous initiation ritual known as "network configuration." Telecommunication software should aggressively try to determine what the proper connection settings are by checking the hardware itself. The best of the current online systems already do this: They begin by checking the modem and printer ports to see if a modem is connected, and, if so, they try to determine which type and choose a settings file appropriate to it. Some even prompt the user for their city, then automatically store the appropriate local telephone numbers for access to the network.

LAN-based services can borrow many of the same tricks. One news reader program I know can be distributed anywhere in the world. When it is first launched, it senses the type of network it is on, and adjusts its protocol and other settings accordingly. It next accesses a master server that tells the program where the nearest local news server is. The program then begins to read news from that local server. The next time the program is launched, it tries that news server first (since it worked last time), and only goes back to pick another if the server is down.

Have your program take its best shot at self-configuring, but allow the user to change the settings if things didn't work out. If the user winds up having to configure the program manually, very little has been lost. If the self-configuration works, however (as it should), you'll have saved the user from a load of anxiety and wasted time. You'll also have rescued your technical support staff from answering the number one network support question: "How do I set this !@#! thing up?"

Shopping the 'Net: Three Types of Navigation

Once you get online, the most obvious problem is navigating—trying to find the things you're looking for. It might help to think of online service users as people shopping for information. The trick in designing a navigation scheme, then, is realizing that people shop in at least three different ways:

- *Targeted browsing.* This is where you choose a category of information, then use various signs and subcategories to help you in your search. Targeted browsing is probably the most common type of shopping, where you're looking for things you don't purchase very frequently. Think of it as the "department store" example. If I were looking for a new set of curtains, I'd probably start by looking on the store directory for the "Linens and Draperies" department. The directory might point me to the fourth floor. Once there, I'd look around manually, noticing the types of goods displayed, eventually heading over to a corner that seemed to have draperies, blinds, and other window coverings. Having narrowed my search further, I might simply browse until I found the exact sort of curtains I was looking for, or I might try to get even closer to my goal—say, to the curtains section, or to the section that featured curtains by a particular maker I was interested in.

Most systems offer some sort of targeted browsing through the various system menus (or icons on AppleLink). The strength of this approach is that you don't need to know exactly what item it is you're looking for—you can start out with a general idea, then narrow your search as you feel you're getting closer. It also gives you the ability to notice related information in the same category that you didn't know about, but might find of interest. If I went into the "Sports" section on America Online to find the latest hockey scores, for example, I might also come across interesting articles on the players' talks or a profile on Wayne Gretzky.

• *Hypertextual browsing.* This is a lot like wandering through a mall or a flea market with no particular purpose in mind—just following whatever interests you at the moment. I unfortunately tend to do most of my shopping this way. I might start at Tower Records picking up some new CDs, when I suddenly recall that my record player is at the repair shop and needs to be picked up. I head over to the shop, after which I realize that my car needs its oil changed. This puts me in mind of an incredibly dull conversation I once had with my brother about the merits of 10 W-30 vs. 10 W-40 motor oil, which in turn reminds me that his birthday is coming up. So after I stop at the oil changer, I head off for the stationery store to get him a card. By this time, I'm halfway across town and have completely forgotten where I've come from. Anyone who's used Mosaic for a few hours to browse the Web knows this feeling all too well.

• *Direct access.* If I need to go someplace often or if time is important, I just want to get where I'm going. For me, an example is going to the post office to mail a package. There's no enjoyment to be had in driving the three-quarters of a mile to get there through eight stop lights. I just want to get there, mail the package, and get back; anything that prevents me from getting there immediately is considered an annoyance. This same need for instantaneous movement is embodied in the various commands that let you jump from department to department in most online systems.

The fact is, online services need to support all these methods. They need targeted browsing to let you find new information in an orderly manner, hypertextual browsing to allow you to follow odd paths to related areas of interest, and direct access to let you quickly go to the areas that are most useful. No single interface, whether it be AppleLink's area icons, the Web's maze of hypertext, or eWorld's "City," can meet all these needs on its own. A great online system should pick one method as a base (usually the targeted browsing method) but allow access to the others.

The Basics Still Apply—They're Just More Advanced

Finally, keep in mind that the fundamental rules of good interface design weren't suspended when a modem was attached to the system. If anything, the core principles, such as consistency and feedback, count more than ever. A network can be a confusing, complicated, and uncertain place. Our applications need to work harder than ever to give the user a sense of comfort and control.

For instance, one basic rule of interface design says that the computer should immediately respond to users' input. Buttons need to highlight instantly, and pop-up menus need to pop up as soon as the mouse button is pressed over them. If as little as two-tenths of a second goes by between a user act and a system reaction, users will begin to lose their sense of control over the system. After only a few more seconds, users will conclude that the system has hung, and will actually press the reset button.

Networks can make responsiveness difficult, but they don't change the user's fundamental need for it. To make up for this, programmers have to use their art to its fullest. Load up lists of common options at program startup so they can be presented in a flash when the user needs them later. Cache the user's last few screens so that they can be drawn more quickly if the user chooses to back up. When the user issues a command that will result in a lengthy network operation, react to the input immediately by changing the cursor or displaying a progress bar before going on to actually process the request.

Use metaphors (such as boxes, folders, or even cities) to help the user map out unfamiliar terrain. Keep the aesthetic integrity of the system together so that users have the same expectations and feel the same sense of comfort they would using a non-network application. Work hard to enforce consistency and network transparency throughout the application. In short, do all the things a good interface does anyway, but do them better.

Building an online system that does all of this isn't easy, but it is possible. And in an area which is still something of a dark alley in the human interface, the first person who shines a light is going to be easily seen.

Till next time,
Doc

AppleLink: THE.DOKTOR

Got an interface gripe? Send it to Doc at AppleLink address THE.DOKTOR or Internet address the.doktor@applelink.apple.com.

OpenDoc

continued from page 15

Instead of parts and part handlers, OLE has objects and servers. But things aren't as neat on the OLE side. The server associated with a given OLE object can be one of three distinct types:

• *A local server.* Also called an *embedded object server* or an *EXE server,* a local server is a Windows-based application (such as Microsoft Excel) that can create OLE objects. Such applications provide maximum functionality but cause noticeable delays when the objects that use them are activated—because of the size of a full-featured application, the switching process can take more than a few seconds.

• *An in-process server.* This is implemented as a dynamic linked library, or DLL—a library of code that the currently running application can dynamically load, use, and unload, as needed. An in-process server switches faster than a local server, but it presents other problems. One such problem comes from its implementation as a DLL, which come in two flavors—16-bit and 32-bit. A 16-bit DLL works with 16-bit operating systems (such as Windows 3.1) but not with 32-bit operating systems (such as Windows 95 and Windows NT); the reverse is true of 32-bit DLLs.

So if you implement an OLE server as either an in-process server or an in-process handler (covered next), you will have to create two different versions, one for Windows 3.1, and another for Windows 95 and Windows NT. (*Note:* OpenDoc parts will have this same problem, since they are also implemented as DLLs.)

• *An in-process handler.* This is also implemented as a DLL. It is smaller than an in-process server, but it can do only a limited number of things. Because of this, it may depend on a default server (which may be a local server) to perform tasks it doesn't know how to do.

OLE Selection and Activation

OLE is troubled by two selection modes that complicate the user experience (and make life harder for developers):

• *Outside-in objects.* To select an outside-in object (an object inside another object), you must double-click it. Furthermore, if another object is embedded inside this object, you must click or double-click (depending on the kind of object it is) the embedded object to select it. If outside-in objects are nested *N* levels deep, the user has to activate *N* objects to select the innermost one—and some applications even force nested objects to open in their own windows. This, folks, is *ugly!*

• *Inside-out objects.* If an object and all the objects inside it are inside-out, you can select any of them with a single click. (This is the model that OpenDoc always uses.) To contrast this with outside-in objects, you can select the innermost of *N* nested inside-out objects with one click.

But, wait, there's more! OLE also has two activation modes. (An OLE object must be activated before you can modify or interact with it.)

• *Edit-in-window activation.* Here, when you activate an object, it appears in its own window, ready for editing. (The horror, the horror!)

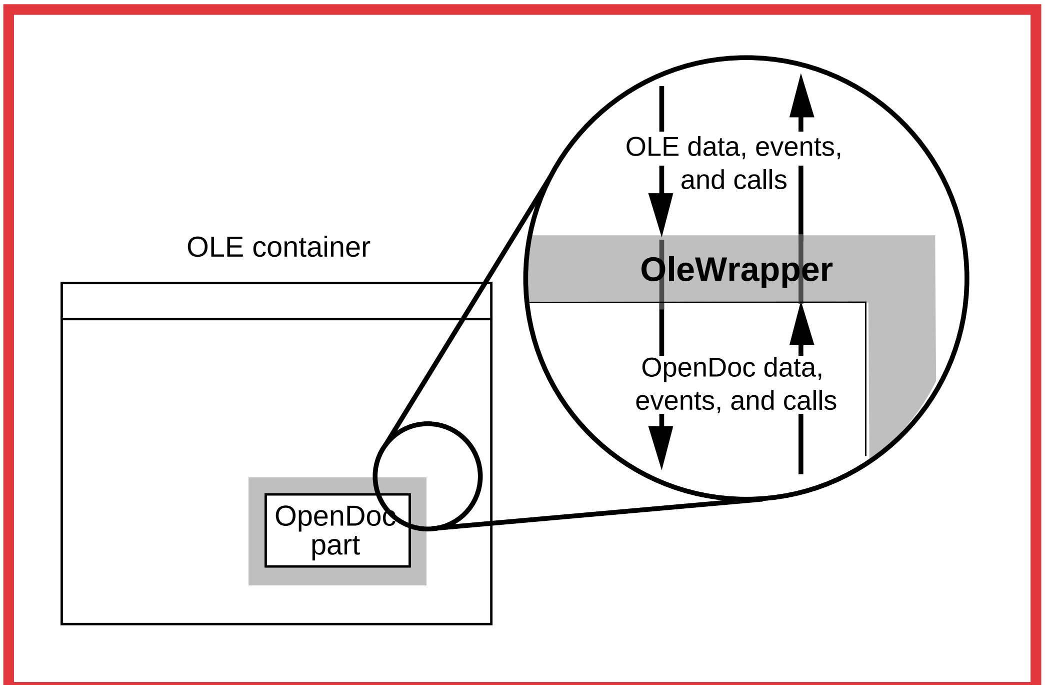
• *In-place activation.* When you click an object that supports in-place activation, it appears ready for editing "as is," apparently inside the document containing it. (I say "apparently" because when the object's server draws it, it actually draws it in its own window, but it draws only the content of the window—no borders or title bar. OLE also incorporates the object's menus into the main menu.) This explains why OLE supports only rectangular objects and can't support features like transparent objects or objects with holes in them.

Put it all together, and you have . . . an unholy mess:

• OLE 1.0 applications support edit-in-window, outside-in objects. Users must double-click them (repeatedly, to get to nested objects), and they appear in their own windows.

• Most current OLE 2.0 applications support in-place, outside-in objects. They appear to be edited in their container window, but users must double-click them (repeatedly, to get to nested objects). Since most OLE 2.0 applications, including Microsoft Excel and Word, don't support inside-out activation, OLE application developers don't have much incentive to do the extra work needed to make their objects support it.

• OLE custom controls (covered later in this article) and OpenDoc both support in-place, inside-out objects. Regardless of the object's nesting, users can edit such objects, seamlessly, with a single mouse click.



The OleWrapper object. This figure shows conceptually what OleWrapper does for an OpenDoc part embedded in an OLE container. The OleWrapper object does not have any "width." The shading here simply denotes that OleWrapper serves as a translator between the OpenDoc world (inside the part) and the OLE world (outside the part).

Oh, By the Way . . .

. . . if you create OpenDoc parts, you don't have to worry about any of this. They'll work in OLE containers at least as well as OLE objects will, and *you* need only support one programming model. Let's take a look at how that works.

How OpenDoc Works Within OLE

On Windows-compatible computers, OpenDoc is implemented as a local server—that is, as a standalone (.EXE) Windows program. An OpenDoc part handler is implemented as a DLL. The OpenDoc server (also called the *OpenDoc shell*) implements a class of OLE objects called an *OleWrapper.*

Imagine an OLE object. It's a rectangular area of content that fits nicely inside an OLE container. Associated with it is a server of some sort—code that executes when the user activates the object.

The OleWrapper code (which is part of the OpenDoc shell) creates just another kind of OLE object. When you click an OleWrapper object to activate it, its server code handles that event. The only thing unusual about OleWrapper is that the only content it knows how to hold is . . . an OpenDoc part!

For a set number of OLE routine calls—including Open, Activate, Deactivate, Hide, Show, and Set Size—OleWrapper receives the call, translates the parameters given, and sends the corresponding OpenDoc message to the appropriate part handler. (See "The OleWrapper object" on page 18.) Instances of the same OleWrapper class are used to "wrap around" different kinds of OpenDoc parts, and OleWrapper knows how to route messages to the correct part handler.

For example, when OLE sends the OLE function DragEnter (during a drag operation) to one of OleWrapper's interfaces, the OleWrapper server code translates that to an ODPart::DragEnter callback on the correct part handler.

(Sometimes, however, there is a substantial difference between the behaviors of OpenDoc and OLE. See the box "OpenDoc Makes Up the Difference" for an example of the kinds of details ComponentGlue takes care of.)

Once an OpenDoc part (wrapped in an OleWrapper object) has been activated, however, the OleWrapper object turns control over to the OpenDoc part handler. This means that, while the OpenDoc part is active, keystrokes and mouse clicks are handled directly by its part handler, *exactly as they would be if the part were running inside an OpenDoc document.* (See "No overhead for an active OpenDoc part" on page 20.) This is why OpenDoc parts will run as well inside an OLE container as they do inside an OpenDoc part.

How OLE Works Within OpenDoc

Now that you've seen how an OpenDoc part works within an OLE container, it doesn't take much to understand how an OLE object works inside an OpenDoc document. The Novell team has implemented an OpenDoc part handler called *OlePart.* This part handler creates what looks to OpenDoc like a simple rectangular part.

To OLE, however, an OlePart part looks like a simple in-place OLE container that holds one OLE 1.0 or 2.0 object. The OlePart part handler implements OLE 2.0's functionality. Because of this, it supports OLE 1.0 objects (but only as well as OLE 2.0 does), both types of objects (inside-out and outside-in), and both activation models (edit-in-window, and edit-in-place). When the object's server is inactive or unavailable, OlePart displays the enclosed OLE object's cached image whenever OpenDoc asks it to draw itself.

When the user double-clicks an OlePart part, OpenDoc transfers control to the server associated with the enclosed OLE object; the server activates the OLE object exactly as it would if it were inside an OLE container. Depending on how the OLE object was implemented, the object may allow itself to be edited in place, or it may open in its own window. Either way, the OLE object is controlled directly by its server; because of this, it incurs no overhead because it's contained by an OpenDoc document.

Support for OLE Custom Controls

You've probably heard of Visual Basic, a "construction kit" that allows users of Windows-compatible computers to create Windows-based applications quickly using a version of the BASIC programming language. One thing that made Visual Basic so successful is that programmers could save thousands of lines of code by dragging predefined Visual Basic custom controls (or VBXs) into a window, thus bringing the functions of those controls into the programmer's application. Another was that programmers could create their own VBXs—which they did, making Visual Basic even more useful.

Unfortunately, VBXs are limited to the Visual Basic and 16-bit Windows 3.1 environments. Microsoft wanted to encourage a

OpenDoc Makes Up the Difference

OpenDoc and OLE 2.0 are similar in many respects, and the OleWrapper server uses the appropriate OpenDoc routines in such a way that the OpenDoc part handler never knows that it's not in an OpenDoc document.

In some cases, OLE contains behaviors that are not needed by OpenDoc, so the Windows implementation of OpenDoc "makes up the difference" by providing the additional code needed to meet OLE's requirements.

For example, in OLE, several cases exist where an object's server is either not running (the object is not active) or not present (the object is in a document on a computer where the server is not present). Because of such cases, OLE caches a representation of each OLE object each time the object is redrawn. That way, if the object receives a "draw" request and the needed drawing code isn't available, the default OLE 2.0 handler services the request by redrawing the "dead" cached representation.

OpenDoc parts don't have this problem—they can draw themselves even if they're not the active part. However, to ensure that OpenDoc parts work correctly inside an OLE container, the local server that implements OpenDoc on the Windows platform caches each part's representation whenever it is redrawn. It does this automatically (saving this representation when the container is saved to disk), and the OpenDoc part does not even know that this is happening. (More important, the OpenDoc part's code doesn't have to be changed to make this caching happen.) Because of this, an OLE container can always draw an embedded OpenDoc part correctly, even if its part handler isn't present.

similar solution that would work in the OLE world (and with Windows 95 and NT) and would form the basis for an eventual move to component software. Microsoft looked at OLE and devised a solution—OLE custom controls (OCXs). An OCX is, essentially, an in-place activated, inside-out OLE object.

Novell is working to make sure that OpenDoc fits into the world of OCXs. Though the work is still in progress, Novell expects that OpenDoc documents will be able to use OCX components and that OpenDoc part handlers can be used as OCXs within OLE containers. You'll see more information on this technology in *Apple Directions* as it becomes available.

More Bang for the Buck

The beauty of the OpenDoc ComponentGlue architecture is that it offers some pretty impressive benefits to both you and your customers:

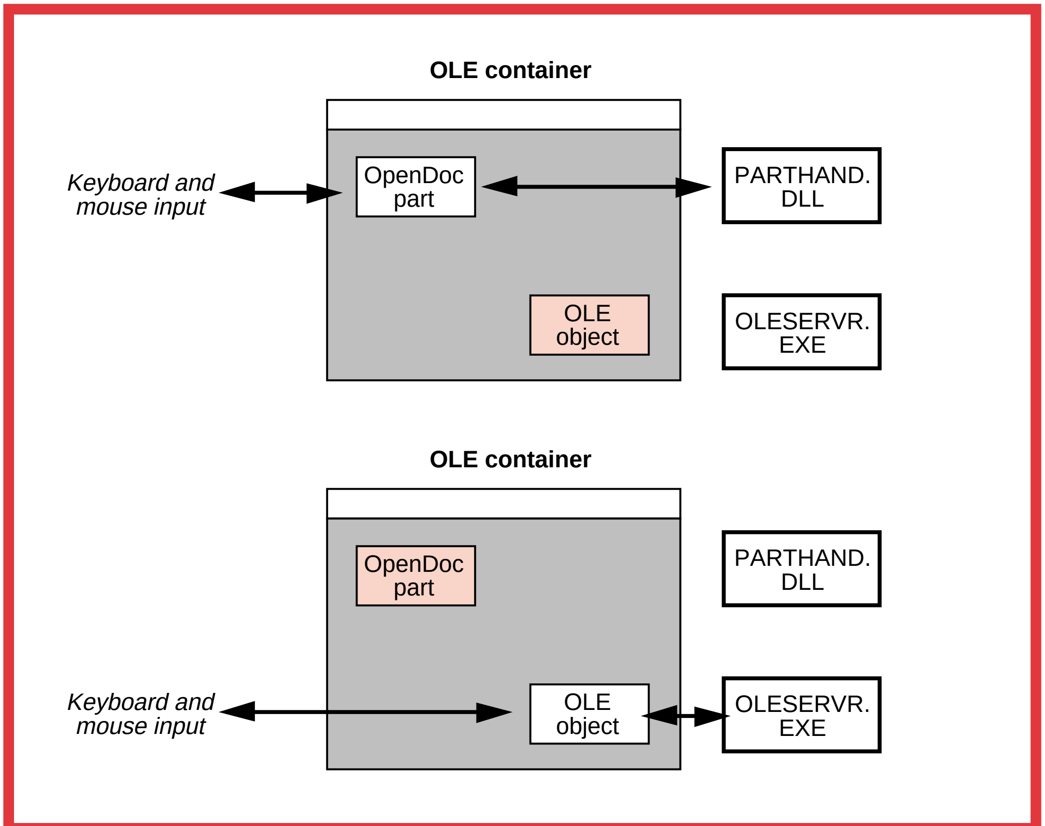
- It allows customers to use OpenDoc and OLE interchangeably, and it protects their investment in OLE-compliant applications.
- Similarly, it doesn't force you to scrap whatever OLE work you've done so far. You can begin to use OpenDoc—which is technologically a better solution—knowing that your OpenDoc parts will work with your OLE objects, and vice versa.
- The market for OpenDoc parts will be as large as that of OLE objects (Microsoft has ported OLE 2.0 to the Macintosh) and will someday be greater—OpenDoc will eventually work on UNIX, OS/2, and perhaps other platforms.
- OpenDoc ComponentGlue allows you to create OLE-compliant software using a much simpler, cleaner architectural model. You won't have to worry about messy OLE implementation details such as multiple activation models or inside-out and outside-in objects.
- By supporting OpenDoc, you don't have to worry about whether you're supporting the "winning" architecture. OpenDoc allows you to support whatever your customers want.

Add to that the advantages of OpenDoc over OLE:

- OpenDoc is a vendor-neutral, open architecture, supported by an independent company, Component Integration (CI) Labs. Source code for OpenDoc is available to any CI Labs member. OLE, on the other hand, is a proprietary architecture controlled by Microsoft.
- OpenDoc is based on a newer, cleaner architecture than OLE. OpenDoc represents more of a break with the architecture of monolithic applications. OLE, on the other hand, is an attempt to squeeze a few more years of life out of monolithic applications by bolting component-software features onto the considerable baggage of today's applications.
- You can create more compelling solutions with OpenDoc parts than with OLE objects because OpenDoc parts can do things OLE objects can't. For example, OpenDoc parts can have multiple frames, while OLE objects are limited to one frame. (Think of a frame as an interpretation of, or "window" into, some or all of a part's data.) So you could have a single OpenDoc part that displays the same data as both a table of numbers and an equivalent bar chart; this would require two separate OLE objects.
- OpenDoc parts can communicate with each other and can change their appearance to reflect changes in the environment, even when they're not the part the user is currently working on. In OLE, only the active object can change its appearance; all the other visible OLE objects are "snapshots" of what they looked like the last instant they were active.

True, Apple and Novell haven't delivered their combined solution yet. But you can be sure that both companies are putting top priority on making sure that the OpenDoc solution meets everyone's expectations. We hope that this article tells you enough about the OpenDoc technology to get you excited about using it—soon. ♣

Thanks to Chris Andrew, project lead of the ComponentGlue project at Novell, Inc., for his explanation of ComponentGlue and his review of this article.



No overhead for an active OpenDoc part. When an OpenDoc part is active (top), it is controlled directly by its part handler, just as an OLE object is controlled directly by its server when it is active (bottom). This means that the ComponentGlue technology forces no overhead on an OpenDoc part just because it's embedded in an OLE container (instead of an OpenDoc document).

CD Highlights

continued from page 14

SCSI Monitor reports aggregate information on disk I/O. It reports the number of disk read and write operations performed, the number of sectors involved, and the time taken to complete the operations.

IOTracer collects trace data on file system and driver read and write operations and other I/O-related calls. We've also provided an analysis tool that inputs an IOTracer trace and outputs a summary of the I/O performance. IOTracer can also collect data on resource calls, which can give insight into the effect of application/system resource requests and disk I/O performance.

These tools can help you understand an application's disk I/O behavior and interaction with the system. They can also help you tune an application's disk I/O performance. A disk I/O test program called BlockReader is also included.

DTS QT Utilities.Apr-95

This is a collection of useful QuickTime functions and applications. For details, consult the documentation in the Documentation folder.

LaserWriter 8 v8.3

This folder contains version 8.3 of the LaserWriter 8 driver and a new version of PrintMonitor.

LaserWriter 8 now provides support for color matching and for embedded JPEG-compressed documents. The driver supports ColorSync 2.0, Apple's latest system software for device-independent color-matching services, and PostScript™ Level 2 color matching.

Desktop PrintMonitor provides a desktop printing environment similar to that found in QuickDraw GX. In this environment, functions previously found through the Chooser and PrintMonitor are accessible on the desktop. Additional features such as print queue management, simultaneous printing to multiple printers, and a drag and drop interface are provided by Desktop PrintMonitor. For details, see the document LaserWriter 8 News.

LockVol 1.6

LockVol is a control panel device that, when used with System 7 and file sharing, can test software handling of locked volumes, local or remote. LockVol displays a pop-up menu containing all volumes in use by your Macintosh and allows you to give an unlocked volume a software or hardware lock. Such changes are immediate and remain intact until you change them back or remount the affected volume. Although the locked or unlocked state of the startup volume is restored after a restart, the states of all other volumes are lost after a restart.

MacsBug 6.5d12

This is the latest nonfinal version of MacsBug, Apple's object-level debugger. See the document MacsBug 6.5d12 Changes for details about this release.

Note: Since this is not an official quality-assured release, it is not guaranteed to work reliably on all hardware configurations. Be aware that there may still be compatibility problems on machines with 68000 processors.

MoreFiles 1.3

MoreFiles is a collection of high-level routines written over the last couple of years to answer File Manager questions received by Apple Developer Technical Support. The routines have been tested (but not stress-tested), documented, and code-reviewed by Developer Technical Support. This release adds new routines and fixes several bugs.

MoreFiles provides

- high-level and FSSpec-style routines for parameter-block-only File Manager calls
- useful utility routines that perform many common operations related to the File Manager
- a robust file copy routine
- a recursive directory copy routine
- catalog-searching routines
- high-level and FSSpec-style routines for Desktop Manager calls
- routines for dealing with pathnames

See the file !MoreFiles Read Me for a description of bug fixes and improvements in version 1.3.

Open Transport 1.0b1c3

This folder contains the Open Transport V1.0b1c3 version of the beta software for 680x0 and Power Macintosh systems. It also includes documentation for all developers and some sample code using this new version.

Pippin Q&A

The 66 Most Frequently Asked Questions Regarding Pippin (and their answers!). *[Editor's note: This document provided source material for last month's* Apple Directions *article "Pippin: A New Platform for Multimedia Titles."]*

QuickDraw 3D ß

This is a pre-release version of QuickDraw 3D for you to investigate. Please note that although we're making this version available to you, it has bugs in it (if it didn't we'd probably be shipping it right now). The documentation is also pre-release and is slightly out of sync with the API.

This version uses 3DMF, a consistent cross-platform (Macintosh, Windows, UNIX®) file format that supports 3D information, from geometry to lighting, textures, and shading. Using the 3DMF file format you can easily communicate 3D information (or its appearance subcomponents) between applications and across platforms. In addition, if your application implements the 3D viewer, you can display and manipulate 3D data from within that application. Finally, by incorporating the full 3D API, any application can offer 3D services. QuickDraw 3D is fully extensible in every aspect of the API.

Implementing 3DMF format import/export capability involves a five-line code addition. Implementing the viewer requires only slightly more effort, equivalent to adding QuickTime support. The API is a more substantial undertaking but offers significant rewards for particular applications.

QuickView Developer Edition

This is the developer edition of the QuickView development environment. You can use this environment to create QuickView databases for your own information that work with the Macintosh Programmer's Toolbox Assistant. Use of this development kit is covered

CD Highlights

continued from page 21

by the license included with this application.

SoundApp

SoundApp is a sample application for demonstrating the Sound Manager. It comes in two parts: SoundApp.c and SoundUnit.c. SoundApp.c handles the application's management of memory, errors, user interface, and so on. It requires the use of the SoundUnit.c file to handle all of the sound routines. The SoundUnit code, which handles the Sound Manager part of this sample application, can be used as a stand-alone unit in your own application.

Universal Interfaces

This folder contains Universal Interface files for Macintosh development.

- *Universal Interfaces 1.0.* These interfaces are the original universal interfaces that were released in the Macintosh on RISC SDK. They had no version number when released, but have now been retroactively named version 1.0. Included in the folder are aliases to other interface files not released with the base set (for example, ColorSync, Apple events, and so on). These other interface files may or may not be universalized.
- *Universal Interfaces 2.0.* These are the final 2.0 interfaces. They were released on E.T.O. #17, in the folder called MPW Latest. The 2.0 release is the beginning of a new release strategy: Instead of each project team at Apple distributing interface files with their own SDK, the project teams will roll their interface files into the next 2.0 release suite. For you, this will mean one-stop shopping for interface files. As you can see, most of the "Other" interface files from version 1.0 have been rolled into the 2.0 release.
- *Universal Interfaces 2.1B1.* These are pre-release updates to the 2.0 interfaces. Version 2.1B1 contains 2.0-compatible C and Pascal interface files for QuickDraw GX, MacTCP, ColorSync 2.0, and AOCE. They were released on E.T.O. #17, in the folder MPW Prerelease. These files are designed to be merged into the Universal Interfaces 2.0 files.

For more details on specific changes, please see the document Toolbox Libraries & Interfaces.

Coming Next Month

What with all the fuss about System 7.5, it's been awhile since we've had a System Software Edition with piles of localized printer drivers, system extensions, and other system software–related packages. I'll see what I can find on Apple's servers. . . .

Alex Dosher
Developer CD Leader

develop *Issue 22*

continued from page 14

asynchronous Apple events.

- "Custom Color Search Procedures" explains a useful method of customizing Color QuickDraw's color handling.

There's also a column on MacsBug for PowerPC, tips for building a shared development environment, and advice on scripting (the latter in a new column by Cal Simone, author of a previous *develop* article on scripting). On the lighter side, you can read about juggling patterns in The Veteran Neophyte or try your hand at the Puzzle Page's latest debugging dilemma.

You'll find all this on the Developer CD, or in printed *develop* if you've subscribed through APDA. Also on the CD are the corresponding source code and two "preliminary" articles: one on multipane dialogs and another on timing. We figure this should keep you busy for awhile—but if there's still something you'd like to see in *develop,* please don't hesitate to let us know at AppleLink DEVELOP.

Caroline Rose
Editor, *develop*

Delta Alert: MPW Pro #17, E.T.O. #17, and Mac OS SDK #3

To help you keep track of improvements in Apple's software tools, here are the highlights of the latest changes to the MPW Pro #17, E.T.O. #17, and Mac OS SDK #3 CDs.

MPW Pro and E.T.O. #17

These CDs should be available by June. They include the following changes:

Latest MPW. Several tools have been promoted from Prerelease to Latest, inlcuding Backup, Choose, Commando, CreateMake, DumpFile, DumpSYM, Pascal, and PPCLink. In addition, Latest MPW now includes version 2.0 of the universal interface files for C and Pascal.

Prerelease MPW. Several new tools, libraries, and interfaces are included in this folder, including late beta releases of MrC/MrCpp (C/C++ compilers that generate PowerPC code) and SC/SCpp (C/C++ compilers that generate 680x0 code). Many of the new tools and libraries are designed to support the new CFM-68K runtime architecture.

Macintosh Programmer's Toolbox Assistant. This is an invaluable new online reference tool to all the Toolbox routines described in *Inside Macintosh* and is designed to replace the MPW 411 help facility. Its built-in hypertext links make browsing and quick navigation easier. You can use MPTA from within either MPW or the Symantec C++ for Power Macintosh 8.0 environment. (See the document Read Me - MPTA for MPW in the Tools for MPW subfolder for instructions on how to use MPTA with MPW.)

Symantec C++ for Power Macintosh 8.0. This is a new C/C++ development environment for Power Macintosh applications. It includes full-featured C and C++ language support, multiple inheritance, and templates. It also supports exceptions through an advanced macro and library implementation, jointly developed by Symantec and Apple. (*Note:* This software is not included with MPW Pro.)

Version 8.0 has many new features: The environment is fully scriptable and recordable; the editor features automatic syntax highlighting, automatic pop-up markers, automatic formatting, and unlimited split panes; and the project manager supports nested projects, named option sets, automatic precompiled header tracking, and drag and drop. Read the User's Guide for all the details.

This software also includes the Symantec C++ for Macintosh development environment for creating 680x0-based applications.

MacApp 3.1.3. This is a maintenance release of MacApp 3.1, including several bug fixes and support for new compilers—MrCpp and SCpp 8.0 (in Prerelease MPW), Symantec C++ for Power Macintosh 8.0, and CodeWarrior 5.x.

MacApp 3.3a1. This is a prerelease of the next major version of MacApp. Features include all the bug fixes and enhancements also made to MacApp 3.1.3; additional bug fixes that involve API changes; performance improvements; support for attaching and executing AppleScript scripts in MacApp applications; improved Drag and Drop support (now fully usable); and internal changes to the PowerTalk mailer support.

Note: To respond to immediate developer requirements, but without sacrificing our alignment with OpenDoc development efforts, we are splitting the previously announced MacApp 3.5 release into two separate MacApp releases—MacApp 3.3 and MacApp 3.5. The first alpha of the MacApp 3.3 release is included here; future E.T.O. releases will include later versions of MacApp 3.3 as well as prerelease versions of MacApp 3.5 with OpenDoc support.)

Ad Lib 2.0.2. This is a revised version of the Ad Lib view editor that supports MacApp 3.3a1.

MacsBug 6.5d12. This new version of MacsBug supports a variety of new features, including greatly improved support for PowerPC debugging. Read the MacsBug 6.5 Release Notes document for all the details.

Power Macintosh Debugger 2.0b3. This is a late beta release of the Power Macintosh Debugger that includes several bug fixes. Note that we've also included a new electronic version of the Macintosh Debugger reference manual.

68K Macintosh Debugger 2.0b1. This is an early beta release of the 68K Macintosh Debugger.

CFM-68K Runtime Enabler. This is the beta release of a new run-time architecture for 680x0-based Macintosh applications and shared libraries, modeled after the Power Macintosh run-time architecture. Be sure to read the documentation located in the Essentials:Programming Documentation:CFM-68K Documentation folder.

System Object Model (SOM). This is a beta release of SOM, an object-oriented programming technology for building, packaging, and manipulating binary class libraries. Look in the Documentation subfolder for information on how to use SOM.

Virtual User 2.1b1. This release of the Virtual User automated testing tool features major speed improvements and several bug fixes as well as a significant online reference addendum to the set of printed manuals that describes this release.

Adobe Acrobat Reader 2.0.1. This is a new electronic documentation viewer from Adobe. To install it on your hard disk, double-click the file AcroRead.mac. Only a few documents on the CD-ROM are currently in Acrobat format, but this number will increase in the coming months.

System 7.5. This is a "net install" configuration of System 7.5, QuickDraw GX 1.1.2, and PowerTalk 1.1.1 as well as the recent System 7.5 Update 1.0. Note that this software is the same as that provided to U.S. Mac OS customers, except that the PowerTalk installer installs a version of PowerTalk that does not use encryption. (This version has the same API as the version that uses encryption, so your development efforts will not be affected.)

Mac OS SDK #3

Below are the changes to the Mac OS SDK #3. This release will be included in the June developer mailing to Apple Partners worldwide. It will also be available directly from APDA by mid-May.

AppleShare API. This folder contains software and information explaining how to make Server Control calls and how to access the UGLibrary from a native Power Macintosh application. We've also updated the C header files so they can be used for creating native applications.

ColorSync 2.0. This is a new release of the ColorSync color-matching software; it replaces the 1.0.5 release.

Installer 4.0.3 (revision 4). This revision includes 25 localized versions of the installer. It also includes a few small fixes to the examples. Read the document Installer 4.0.3 SDK Rev History for all the details.

Interface Files. These are the "latest" Toolbox interfaces from MPW Pro #17 as well as the Interface.o (680x0) and InterfaceLib.xcoff (PowerPC) libraries. The interface files should also work, with little or no modification, with the Symantec C++ for Macintosh and Metrowerks CodeWarrior environments. Note that interface files have gone through significant changes in the past year or so and some sample code has not yet caught up—please bear with us during the transition.

Macintosh Drag and Drop. Two interesting new pieces of sample code are now part of the Drag and Drop package—FinderDrag and DragsAndLists. Refer to the README documents in the sample code folders (inside Demo Applications) for all the details.

MacODBC. Extensive electronic technical documentation is now included with the MacODBC run-time software.

QuickDraw GX 1.1.1. The QuickDraw GX SDK has been updated and revised to reflect the capabilities of the version 1.1.1 release. Note that many of the Read Me files in this folder are portable digital documents and you will need to install QuickDraw GX to view them—the icon for these files is a newspaper with a handle.

QuickTime 2.0. We've added a file called QT_MAC.PDF (in the Programming Stuff:Documentation folder) that describes all the changes to the QuickTime Toolbox that were made between versions 1.6.x and 2.0. Note that this file is in Adobe Acrobat format, so you will need to install the Acrobat Reader first.

The Mac OS SDK also includes Japanese resources for Apple Guide 1.2.5, a big-fix version of Apple Shared Library Manager 2.0b13, updated information for developers of PCI cards and drivers, and a beta release of Open Transport. ♣

Business & Marketing

Market Research Monthly

The Most Popular Application Categories

Inside This Section

Macintosh customers appear to be taking their computers more seriously than Windows customers. That's one conclusion that can be drawn from the latest study conducted by Apple Computer, Inc., of how people around the globe use personal computers. The results of the study indicate that there remains a considerable opportunity to "lighten up" the Macintosh platform with more games, and the results also confirm several long-held industry beliefs about the platform.

Apple's research is part of its latest User Profile Study, in which the Customer Research department studied customers who used the Mac OS, MS-DOS, or Windows operating systems as of June 1994. These customers were asked to indicate which types of applications they use; the types most popular among the study participants are shown in the box "Top Five Application Categories by Platform" on this page. More complete numerical results appear in the chart "How Personal Computer Customers Use Their Computers" on page 25.

The Case for Macintosh Games

Interestingly, more Mac OS customers—59 percent of them—use their computers for spreadsheet software than for games and entertainment, despite the apparent trend of entertainment quickly becoming the most in-demand software category (and despite the Macintosh computer's traditional lighthearted image). According to the Software Publishers Association (SPA), entertainment was the largest software category—and among the fastest growing—in North America in 1994. (See IndustryWatch on page 3 for more information about the recently released SPA data.) This gap between customers' desire for games and their current use of them suggests there's unmet demand for Macintosh games, a need some of you might want to get busy filling.

Macintosh Dominates Publishing

One of the industry beliefs about the Macintosh platform is that it remains strongest in the desktop publishing (DTP) and graphics categories. Apple's study confirms that belief, showing that nearly half of Mac OS customers use their systems for DTP and graphics. By comparison, only a third of Windows customers use DTP and graphics software. This gives us all the further opportunity to cement the Macintosh computer's lead in these areas, especially now that Power Macintosh hardware can run "native" RISC publishing and graphics software with so much speed. In other words, you'll want to optimize your software in these categories for Power Macintosh systems, if you haven't already.

Digging more deeply into Apple's data about DTP and graphics usage, it's apparent that those who use the Macintosh computer for these tasks tend to use more high-end software, while DOS- and Windows-based publishing and graphics packages are oriented toward low-end or casual usage. Also, the Macintosh business market is heavily graphics-oriented: One in five Macintosh business customers is employed in the graphic design or in-house publishing department of an organization. And one in three Macintosh business customers uses a DTP application as part of his or her main job. In contrast, among Windows and DOS customers combined, one out of ten is employed in a graphic design profession.

Opportunity for Macintosh Financial Products?

Just as the Macintosh platform has traditional strengths in desktop publishing, the Apple study also confirms that DOS and Windows are stronger in the accounting software

Top Five Application Categories by Platform

Here is what Mac OS, MS-DOS, and Windows customers participating in Apple's 1994 Worldwide User Profile Study said were the five types of applications they use the most.

Mac OS

1: Word processing
2: Spreadsheet
3, 4: Database
Games (tie)
5: Desktop publishing

MS-DOS

1: Word processing
2: Spreadsheet
3: Database
4: Games/Entertainment
5: Accounting/Finance

Windows

1: Word processing
2: Games/Entertainment
3: Spreadsheet
4: Database
5: Accounting/Finance

Source: Apple Computer, Inc. © 1995

category. Fully half of DOS and Windows customers use accounting software, compared to just over one-third of Mac OS customers. We think this spells another opportunity: for new Macintosh accounting software.

Here's why: Last November, this column reported that sales of computers to customers in the small office, home office (SOHO) market can be expected to grow quickly in the mid-1990s. (See "Surge in Small Office, Home Office Market to Continue" on page 21 of the November 1994 issue.) Apple data on the SOHO market indicates that accounting software is one of the top four application categories in that market. The expected growth in the SOHO market, and the importance of accounting software for it, points to an opportunity for new, high-quality home financial software (especially if it can take advantage of native Power Macintosh performance and the floating-point unit).

Before closing, a caveat: As we've said before, we can only point the way to opportunities; we'd strongly advise you to conduct your own carefully targeted customer research before pursuing any of them. ♣

How Personal Computer Customers Use Their Computers, 1994

This chart shows the percentages of Mac OS, DOS, and Windows customers who use each category of software.

	Mac OS	MS-DOS	Windows
Word processing	94%	88%	93%
Spreadsheet	59%	61%	63%
Database	49%	53%	52%
Games/Entertainment	49%	51%	78%
DTP	48%	20%	33%
Drawing/Painting	45%	17%	33%
Accounting/Finance	38%	48%	50%
Calendar	38%	33%	40%
Education	26%	24%	31%
E-mail	25%	25%	26%
Online services	22%	22%	29%
Presentation	20%	12%	24%
Engineering/Science	12%	8%	16%
Programming	10%	12%	14%
Multimedia	10%	3%	15%

Source: Apple Computer, Inc. © 1995

Special Marketing Report

Staking Your Claim on the Internet Frontier

By Kevin Ohlson,
Ohlson Consulting

The Internet, unknown to all but the lunatic fringe of computer users just a short time ago, is growing at a rate that defies description. The media has helped fuel this growth, mentioning the Internet in 35,619 stories in 1994, and an estimated 70,500 stories in 1995 (source: Dow Jones News Retrieval). Computer columnists, media pundits, analysts, and prognosticators would have us believe that the Internet represents a bonanza unparalleled since the Oklahoma Land Rush. Yet, even after stripping away all the hype, there is real business opportunity for those in the software development community.

This article outlines some Internet opportunities and discusses the issues surrounding them. (See the article "Programmer, Get Thee to the Internet!" on page 29 for details on how to get connected to the Internet and where to find Macintosh programming resources.)

A Brief History of the Internet

First, some background on the Internet. Descended from a DARPA (Defense Advanced Research Projects Agency) proposal for a packet switched network in the late 1960s, Internet TCP/IP protocols were developed at Stanford University, University College London, and BBN (Bolt, Beranek, and Newman) labs. In the ensuing 20 years, the Internet became an international communication web among academic and government research facilities. And to that end, a number of tools and applications were designed that enabled file transfer, remote computer use, electronic mail, and discussion group communication. By the late 1980s, efforts to coordinate commercial activities with Internet resources began to take hold, resulting in further increases in users and connected networks.

In early 1993, Tim Berners-Lee at the European Particle Physics Laboratory released a specification for a client/server architecture that facilitated the development and distribution of research papers. This specification became the basis for the World Wide Web (WWW), a unified "information space" that consists of hypertext documents and links between documents. What made this innovative data and addressing standard revolutionary was its openness, independence from hardware or software, and flexibility. All of a sudden information providers could create graphic front ends for an assortment of data types, including text, graphics, sound, video, and hypertext links. And, most important, users could easily access this information from anywhere in the world, with virtually any computer system, a capability that helped Internet awareness and usage grow exponentially.

How Big Is the Internet?

How big is the Internet, and who's connected? The original Internet was designed to accommodate up to 256 research nodes, which in 1973 probably seemed more than adequate. Today, according to the Internet Society, there are over 50,000 networks and 4 million hosts. More than 90 countries are connected directly to the NSFnet (National Science Foundation Network) backbone, and more than 150 countries have e-mail access. And it's been reported that as many as 50 million users have, at a minimum, e-mail connectivity "to the net."

In October 1994, Texas Internet Consulting estimated that approximately 14 million people had full Internet capabilities. Today, the number is probably closer to 20 million users, as individuals, businesses, and commercial online services such as Prodigy and America Online begin accessing the Internet. And given the current growth rate of 10 percent per month, some forecasters predict, with tongue in cheek, that Internet use will exceed the human population by the year 2000.

Formerly the domain of academia and government-sponsored research organizations, the Internet is now being used by corporations and individuals. Commercial networks (those ending in ".com") recently became the most prevalent type of network, surpassing educational, government, and not-for-profit networks. The Internet Society estimates that 66 percent of the Fortune 500 companies have some Internet presence. It's also projected that non–U. S. networks will come into a majority in 1995, reflecting the Internet's international makeup. And the Internet is rapidly reaching the consumer market. Find/SVP, a primary research operation in New York City, says that over 3 million households in America have Internet access beyond e-mail.

In contrast, findings from Apple Computer's 1994 U.S. Macintosh User Profile Study show that only about six percent of Macintosh and Windows customers have used the Internet. One of the reasons this connection rate is so low is because of how difficult it is to connect to and navigate around the Internet. The pent-up demand presents you with opportunities to develop products that enable easier Internet access.

The WWW Publishing Locomotive

In the two years since the World Wide Web began, nearly 30,000 WWW servers and 1 million active users have connected. The reason behind this phenomenal growth is the Mosaic graphical browser. This visually appealing browser enables users to review electronic information using point-and-click operations. And its ease of use enables first-time, nontechnical users to easily access the vast resources on the Internet. Mosaic's intuitive graphical interface, coupled with Web server software, has created a new medium for publishing information.

Traditional publishers, such as Time Warner, CMP Publications, International Data Group (IDG), and hundreds of others, have begun experimenting with the Web, posting publications in Mosaic "Web page" formats. Information included in their print editions, and in some cases more, can be searched and browsed. Government organizations, including Congress, the White House, and many state and local governments have also put up Web home pages. And corporations are using the Web to make product information and press releases accessible to Internet users.

The Internet Gold Rush

Publishing is merely the first phase of commercial interest in the WWW. The prospect of using it for commerce—that is, the direct delivery of goods and services in exchange for money—is really what this Internet "gold rush" is all about. While there are some companies transacting business over the Internet, notably the Internet Shopping Network, most operations on the Internet are just testing the waters and learning more about the technology. They're also trying to figure out how this new medium will change human and organizational behavior. Using data from the Interactive Services Association, I estimate that in 1994 less than $100 million in transactions occurred on the Internet. By comparison, online services such as eWorld, America Online, and CompuServe generated over $400 million in nonsubscription revenues during the same time period.

That $100 million is merely rounding error in the world of commerce. Last year, catalog publishers posted over $53 billion in revenues, and direct marketing operations tallied up well over $400 billion. The Internet has a ways to go before becoming a commercially viable means of commerce. In the meantime, developers are scrambling to overcome some of the problems involved in doing business on the Internet, while seeking new ways to deliver products and services to a "wired" consumer audience.

Internet Opportunities

What's so compelling about the Internet today? After all, it's been around for two decades—why all the recent attention? One explanation is that businesses are just now beginning to understand that "internetworking"—computer-aided communication between enterprises—offers benefits as dramatic as those that accompanied workgroup networking. And the pervasive coverage of these benefits by both the popular and technical media is reinforcing this belief.

The good news for developers is that this corporate awareness is creating a need for better Internet-related applications, products, services, and technologies. I believe the best way to fully take advantage of this momentum is to examine how businesses can use the Internet to increase opportunity and decrease costs.

As developers, you're also in the best position to use the Internet to market your own products and create solutions that help others do the same. At the highest level, I believe the most promising areas for employing Internet technologies are global marketing, distributed collaboration, better links between suppliers and customers, and cost savings. Now I'll explain each opportunity in detail.

Opportunity #1: Global Marketing

At the foundation of any online service is a sense of community. That is, the service brings together people who share common interests. The Internet, with its newsgroup access, electronic mail, and World Wide Web, facilitates a sense of community on a global scale. Everyone with Internet access is only a few mouse clicks away from any other person, product, or service. It's as easy, *and inexpensive,* to communicate with someone across town as it is between continents. The normal geographical constraints of information distribution and interest generation are removed.

So, developing an international presence no longer requires the resources it once did. Many companies are setting up Internet "field offices," using the

Internet to provide international customers with pre-sales information, product ordering, and post-sales support. For example, Digital Equipment Corporation uses the Internet to provide their worldwide customers with access to an Alpha server for benchmark testing. And even small developers are able to conduct business internationally through the Internet.

"About 40 percent of our business is international, and almost all of it comes through the Internet," said one executive at a small software company, who requested anonymity because he considers his company's use of the Internet to be a competitive advantage. "Also, the Internet is a very level playing field, at least at this point. A company can create its own image, regardless of its actual size."

As companies begin to embrace the Internet as part of their overall marketing efforts, they're learning that there's more to being a member of a "global village" than connectivity. For example, 3Com Corporation displays "welcome" messages on its Web home pages in multiple languages, and Netscape Communications posts information in English and Japanese. Yet some basic issues remain. The world is still made up of many different computing platforms, and there are no single standard formats for text, audio, and video data.

Opportunity #2: Distributed Collaboration

Companies are finally getting used to the idea that collaboration does not necessarily require co-location. In fact, many progressive organizations have figured out how to leverage distributed resources, not just tolerate them. This is the idea explored in the best-selling book *The Virtual Corporation.* It puts forth the premise that the best people for any given task are probably not sitting in your department, and that project teams can effectively include experts, full-time employees, and contractors located at different sites.

In many cases, productivity can be increased by enabling telecommuting. As more people and companies rely on "distance working," communications firms are scrambling to provide improved infrastructure through satellites, cable TV wiring, and telephone lines. And workgroup applications and environments are still far from optimal. For example, Lotus Notes, the leading collaboration product, is still primarily geared toward local area network (LAN) environments, not wide area networks (WANs) like the Internet.

In the late 1800s, America's transcontinental railroad tied two halves of a country together into the *United* States, significantly improving commerce and the feeling of community among its people. The invisible tracks of the Internet will, to some extent, bring the world together in the same way.

Distributed collaboration is possible today, but it's not recommended for the technically timid. As developers, you have the power to make this way of working commonplace and effective.

Opportunity #3: Better Links Between Suppliers and Customers

If the age of internetworking is fast upon us, why is it still so hard to send e-mail and attachments? With so many disparate electronic communications systems, including PROFS, cc:Mail, Microsoft Mail, and so on, only the Internet has emerged as the common path between all points. Virtually every electronic mail system has an Internet gateway. So what better system is there on which to base one's communications architecture?

In the 1980s businesses created information infrastructures internal to their enterprises. Electronic "castle walls" were erected around companies, and external networking and electronic communication were either discouraged or prohibited. Today, however, successful businesses rely on close communication with customers and suppliers, increasing their responsiveness. A "castle wall" is still needed, but well-defined points of entry can increase overall customer satisfaction.

The Internet is open, easily extensible, inexpensive, and available to suppliers and customers throughout the world. And the medium is not just limited to static text and graphics—companies are exploring interactive voice and video capabilities, as well. But the administration and use of Internet communication is still too difficult, and providing seamless interfaces for all forms of communication between different proprietary systems is a need that developers can fill.

Opportunity #4: Cost Savings

Today many companies are using the Internet as a means of lowering their cost of doing business. While it's difficult to quantify how much has been saved, hundreds of organizations can point to the Internet as the foundation for big savings. Apple Computer, Inc., is among these companies, as well. Other computer companies using the Internet include the following:

- Sun Microsystems uses Internet technology to distribute press information to the media and analysts, foregoing traditional fax and paper mail. Not only do recipients get information in near real-time and at substantially lower costs to Sun, but they also receive it in a much more useful format than hard copy. The information can be saved, forwarded, or discarded quickly and easily.
- Digital Equipment Corporation was among the first companies to use the Internet to publish marketing collateral information, such as data sheets and white papers. The advantages are compelling: Information is always up-to-date, and the company incurs essentially no distribution or production costs. Digital estimates that this helps them save the cost of printing and distributing between 10,000 and 20,000 paper pieces per month.
- Cisco Systems is using their World Wide Web site to lower support costs to their channel by providing information, product updates, and technical support patches in special sections for those with support contracts. Rather than navigating through an automated voice mail system or waiting for a sales person to return a call, users of the Cisco home page can help themselves to the information that they need. Cisco lowers costs by placing online answers to frequently asked questions, and by using search utilities to help visitors find the information they're looking for. Users can also shop on Cisco's online product catalog, though they're not yet able to place orders. And, of course,

Cisco saves costs by not mailing out as many paper documents, software patches, or updates.

Many companies are working to use the Internet to distribute complete products and documentation. For example, Oracle Corporation has announced their intent to use the Internet as a distribution channel for certain database products. Other companies are exploring the Internet as a connection mechanism to make end-users aware of software updates.

What drives many of these cost reductions is the relatively low cost of getting connected to the Internet. Dial-up accounts can be obtained for as little as $20 per month for ad hoc 14.4-Kbps access (enabling data transfer six times faster than a 2400 baud modem). And full-time 56-Kbps frame relay connections cost under $500 per month. For most businesses, this represents a significant reduction to their existing costs for data connections, postage and printing, and technical support media. And this is perhaps the hidden feature of the Internet: You pay for your own connection costs and capabilities. If the demands of one site necessitate higher bandwidth, such as a 1.544-Mbps T-1 connection, you don't pay for it; the site does. Businesses benefit from global connectivity at local access costs.

The Internet Users' Wish List

What are the product development opportunities for developers? As I stated at the outset of the article, there is much to be done before the Internet is widely used and accepted as a medium for electronic commerce. Below, I've listed needed products and technologies that will help this happen. In some cases no technology currently exists; in others, significant improvement is needed over current solutions. No doubt there are many more.

• *Authoring tools.* The Internet presents developers with an opportunity to make online publishing easier, much as developers improved desktop publishing a decade ago. While less than 6 percent of all personal computer users have Internet access, this number is expected to grow dramatically in the next few years. So there will be a healthy market for those who figure out ways to ease the publishing process and enhance the appearance and effectiveness of published information.

• *Conversion tools.* Most businesses have plenty of existing documentation in digital format, and none are excited by the prospect of having to reprocess this material for online posting. Tools that help users and enterprises convert their existing work into Internet-ready formats will be welcomed with open arms. Better yet, figure out a way for users to create documents in one format that can be used on both CD-ROM and the Internet.

• *Discovery.* With tens of thousands of Internet sites, how is a user to know where to go? And with new sites popping up every minute, how can they keep track of new information? How about an Internet Yellow Pages? There are ways to search the Internet manually, but the best approach is to create intelligent network agents that search for and deliver specific information directly to users. So far, agents have yet to live up to the their promise.

• *Information management.* The Internet was originally designed to facilitate the sharing of information between researchers, and this group of users has data management needs that are significantly different from those of most businesses. An online relational database system, preferably one that handles multiple data types and interfaces to HTML (Hypertext Markup Language) and http (hypertext transport protocol), would be a boon to both businesses and users. What is needed is a World Wide Web database interface with an extension that enables the creation of custom databases.

• *Interactivity.* In a sense, the Internet is a not a very interactive place. You can browse for information, and if you like it, obtain a copy. But what people really want from connectivity is interactivity. Pioneering work is being done on interactive audio ("Internet telephone") and video ("Internet television"). Even further out is the concept of dynamic clients/servers, which would enable users to dynamically download applications that provide them with access to remote servers. With this technology, users could tap into the power of a remote supercomputer with an easy-to-use graphical front end.

• *Digital money.* Although you can buy things over the Internet using a credit card number, the lack of security on the Internet makes it a little risky. Some organizations are working on the creation of reliable "digital money" that could be used to purchase items online. Some ideas in the works involve point-to-point encryption of your credit card number, while other methods try to ensure authentication—in other words, verifying the identity of the purchaser—before an order is completed. So far, none of the proposed digital money solutions are both safe and convenient enough for all concerned.

• *Network devices.* Some Internet locations are accessed over 3 million times per day—that's an average of 350 network requests per second. As you might guess, most off-the-shelf computer platforms need to be customized to work at this rate. In the next few years, there will be demand for specialized high-capacity network servers. On a slightly less demanding scale, providing digital communication products such as higher speed modems and access software to homes and small businesses represents an enormous opportunity. While many think ISDN will provide a solution here, these systems are currently too expensive and difficult to install.

In the late 1800s, America's transcontinental railroad tied two halves of a country together into the *United* States, significantly improving commerce and the feeling of community among its people. The invisible tracks of the Internet will, to some extent, bring the world together in the same way. But with the opening of any new territory, there's opportunity *and* chaos. For businesses and consumers, the Internet represents a dynamic change in the way people interact with each other. So if you're up for a little adventure, hitch your company up to the Internet, and stake your claim on the Internet frontier. ♣

Kevin Ohlson (kohlson@eworld) is a principal at Ohlson Consulting, a consultancy focused on online services. Kevin is also the publisher of the Business Guide to Online Services.

Special Marketing Report

Programmer, Get Thee to the Internet!

By Kevin M. Savetz

The Internet never sleeps. The world's largest computer network hums away 24 hours a day—which is good to know if you're a programmer, wide awake and miserable at 3 A.M. because your code won't compile for some unfathomable reason. The Internet is rife with experts in every imaginable field—and as you might expect, it's full of resources for Macintosh programmers: databases of information, source code snippets, and expert advice.

Every day, some 20 million people in 150 countries access the Internet to exchange electronic mail, information, and files. It doesn't matter whether your signature is emblazoned inside the earliest Macintosh computer or you're just getting past programs that print "Hello, World!" on the screen—you can get help on the Internet, even at 3 A.M.

There are a multitude of services you can use to access the Internet. Some connections give you a wide variety of Internet services and tools; others limit you to relatively few. Many Internet service providers do not offer full access to the Internet's functions. The most basic level of Internet access is electronic mail, which enables you to exchange e-mail with users on the Internet and other networks.

A more complete Internet connection includes access to Usenet (also known as *newsgroups* or *netnews*)—the "bulletin board" of the Internet, where you can read and post messages on any of 9,000-plus topics.

And an even better Internet connection features still more tools—programs that enable you to connect with other people and computers in "real time." These include Telnet, FTP (File Transfer Protocol), and Gopher. The brass ring of Internet tools is the World Wide Web—this is the tool that provides a snazzy hypertext graphical interface.

If you're a "newbie" (a person new to the Internet), read on for ideas on how to get connected. If you're already online, jump to the section "Programmer Resources on the World Wide Web" on page 30 for pointers to some of the best Internet resources for programmers.

Getting Connected—For "Newbies" Only

Before you can explore the Internet, you have to "plug in." That is, you must have access to a computer that is connected to the network. When you buy a telephone, it doesn't work right out of the box; before you make that first call, you must plug your phone into a telephone line and hear a dial tone. Similarly, you can't dial up the Internet's services until your modem can connect with a computer that's part of the Internet. Once you have an "Internet dial tone," you can access the Internet's resources.

Because the Internet is a cooperative effort, there is no "Internet, Inc." to sign up with and send a check to. Instead, you must find a service that is plugged into the Internet. Not every online service is part of the Internet, and, as you will see, the tools available at various services differ considerably. The Internet "dial tone" can take many forms, so there are many choices and features to consider.

Your computer gains access to the Internet through a computer called a *host.* The company or institution that operates the host is called an *Internet service provider* (or *ISP*). Because of the array of computers and people that make up the Internet, service providers range from billion-dollar commercial online services to tiny bulletin board systems running out of someone's basement. Regardless of where you're connected, your service provider is the person or company on the other side of your Internet link.

Getting telephone service is simple and decision-free; you ask the local phone company for a line and you get it. But getting an Internet "dial tone" isn't as straightforward—you have to choose your access method, think about what services you'll use, compare prices, and, finally, sign up with a service provider. Because you connect to the Internet with a phone call, you probably want an Internet service provider that has a phone number in your area, lest you be forced to pay the phone company steep long-distance charges. In the past year or two, Internet service providers have been popping up like weeds, and it has become much easier to find a local "on ramp" to the Internet, even if you don't live in a metropolis.

Individuals and small businesses can best access the Internet using a dial-up connection—which means that your modem dials a host computer to access the Internet. Your phone line is tied up only while you're using the Internet. When you're done, you hang up the modem and free up the phone line. All you need is a computer, a modem, a plain old phone line, an account with a service provider, and Internet interface software such as Gopher or Telnet.

If you already know someone who is part of the Internet community, ask that person how to get access. If that person lives near you and is happy with the service, chances are that service will be right for you, too.

Besides the tools available to you, the service provider you choose determines the type of interface you see once you're online. The interface determines how the Internet appears to you (as simple text on the screen, as a menu interface, or as a graphic screen with sound).

Command-Line Access for Simple Needs

Command-line access through a local Internet service provider is cost-effective, simple to learn, and similar across different computing platforms. This kind of access is easy to use and, unlike Internet Protocol (IP) links, which are described more fully in the next section, command-line access doesn't require complicated software configuration on your own computer.

Command-line access is easy to set up; it is generally a little less expensive than other types of Internet connections. Command-line access works reliably from any kind of personal computer because specialized software isn't needed.

Dial-Up IP for Direct Internet Access

Dial-up IP links such as Serial Line Internet Protocol (SLIP) and Point-to-Point Protocol (PPP) connect your computer directly to the Internet while you're dialed in. You can run networking applications right from your own computer instead of relying on another host. IP links are more robust than other types of accounts—for example, they enable you to connect to multiple sites simultaneously. You can have an FTP session in one window, Telnet in another, and Gopher in yet another. With an IP link, you're generally not married to any particular software. If you don't like the software you're using for e-mail (or for reading Usenet news, or whatever), you can switch to something else.

Because the software for a dial-up IP link resides in your computer, you must find it and install it yourself. You also have to configure many pieces of software on your computer and take complex steps that command-line users need not worry about. The software you need includes several programs: one each for e-mail, FTP, Telnet, Gopher, World Wide Web, and so on. If you're new to the Internet, setting up this software can be a frustrating welcome to the "information superhighway." However, once everything's set up the first time, IP links are easier to use than command-line accounts.

Finding a public access site for a command-line account or dial-up IP link is usually more difficult than joining a commercial online service. Although there are only a few commercial online services, there are hundreds of service providers offering command-line accounts. Each one seems to offer different features, pricing structures, and local access locations.

Commercial Online Services for Multipurpose Access

Unlike providers that offer only a link to the Internet, commercial online services, such as CompuServe, eWorld, America Online, and Delphi, offer a variety of services other than Internet access. These services include databases, online shopping, games, and file libraries.

Every major online service offers some degree of Internet access, though they don't offer every tool. For example, most don't have World Wide Web browsers (though they're certainly coming). And commercial online services are clamoring to provide the "whole enchilada" of Internet tools, which I suspect they all will, very soon.

Get Thee to the Library!

There is an endless variety of books available that can help you get connected and learn your way around the Internet. Here are some of my favorite books for Internet "newbies."

• *Connecting to the Internet* by Susan Estrada. O'Reilly & Associates, 170 pages, $15.95. ISBN: 1-56592-061-9.

This small book focuses on choosing the best type of network connection for your personal, school, or business needs, and how to get the best price for the type of access you require. It explains the differences between SLIP, PPP, command-line accounts, and other options. It includes an extensive list of Internet service providers. This is a single-purpose book, telling how to choose a connection and get online; it doesn't try to teach you how to use the Internet once you've connected.

• *Internet Starter Kit for Macintosh* by Adam Engst. Hayden Books, 640 pages, $29.95. ISBN: 1-56830-111-1.

If you use a Macintosh and only have $30 in the world to spend, buy this book. This is simply the best Internet book for Macintosh users that was ever—or could ever be—written. It claims to be "everything you need to get on the Internet" and it delivers: everything from the software you'll need to get connected to 1,000 pages of Internet knowledge. This is an excellent, enlightening, fun book. If you own a Macintosh and are on the Internet (or want to connect), this is the one to get.

• *The Whole Internet User's Guide and Catalog* by Ed Krol. O'Reilly & Associates, 572 pages, $24.95. ISBN: 1-56592-063-5.

This book covers the basic utilities used for accessing the network and then guides users through the Internet's millions of files. It includes a resource index that covers a broad selection of approximately 300 important resources available on the Internet. The second edition has been completely updated to reflect the development of new Internet tools, including Mosaic, MIME, tin, pine, xarchie, and a greatly expanded resource catalog. I highly recommend this book.

• *Your Internet Consultant—The FAQs of Life Online* by Kevin M. Savetz. Sams Publishing, 550 pages, $25 ($35.95 Canada). ISBN: 0-672-30520-8.

This book (written by yours truly) provides simple, enlightening answers to hundreds (361, to be exact) of frequently asked questions about the Internet, as well as answers to a few questions that aren't asked frequently, but should be. The book is arranged in a question-and-answer format, making it blissfully simple to find just the information you need. I think you'll find the book unique, useful, and a little silly.

Commercial online services are almost always more expensive than IP links and they may offer features you don't need—especially if all you want is a pipeline to the Internet. But a notable advantage to commercial online services is that, unlike most public access providers, they can be accessed with a local phone call from hundreds of cities through packet-switching networks such as SprintNet and Tymnet.

Your choice of online services depends on what type of services you want. Delphi was the first nationwide service to provide full Internet access. Its text-only command-line interface won't win any beauty contests, but it allows you to find your way around easily enough. America Online (AOL) and eWorld offer snazzy graphical interfaces. As of this writing, AOL provides e-mail, Gopher, FTP, and WAIS access. eWorld has lagged behind, offering only e-mail access, but other Internet tools should be available very soon. (However, eWorld has other advantages: It's home to lots of source code, databases, and other developer information that isn't available on the other online services. eWorld is, of course, owned by Apple Computer, Inc., and it sometimes pays to get your information straight from the source.)

AOL's voice information line is 800-827-6364 or 703-448-8700. eWorld's information line is 800-775-4556 or 408-974-1236. For information about Delphi, call 800-695-4005 or send a fax to 617-476-9600.

Programmer Resources on the World Wide Web

The World Wide Web (or WWW) is an easy-to-use, graphical hypertext resource that makes navigating the Internet simple. Not all Internet connections have Web access, but its ease of use and incredible popularity are making it a staple of Internet connections. Here's a sample of the good stuff programmers can find on the Web:

- *Apple Computer's Web Page.* Apple's Web resource is a gem, offering enough consumer and technical information to keep you busy into the wee hours. Developers will revel in online versions of *develop* (the Apple Technical Journal), *Apple Directions,* a complete online database of Macintosh technical notes and developer notes, sample code and sample applications, Apple peripherals documentation, system software, and more. You'll find this programmers' paradise at http://www.info.apple.com/, and it's worth the price of an Internet connection all by itself.
- *Information about MetroWorks CodeWarrior.* You'll find this information at http://power.globalnews.com/articles/4992.htm.
- *Macintosh WWW Page Development Resources.* This page, which is located at http://www.uwtc.washington.edu/Computing/WWW/Macintosh.html, includes a listing of resources that aid the development of WWW servers and data on Macintosh systems.
- *MacHTTP information.* If you're thinking of providing information on the World Wide Web yourself, MacHTTP is the program you need. MacHTTP is a fast and full-featured HTTP server that runs on the Mac. This information is available at http://www.biap.com/.

Usenet—The Internet Bulletin Board

Usenet is the world's largest distributed bulletin board system, shared by millions of people swimming the seas of the Internet. Folks on Usenet talk about everything—everything!—you can think of, from square dancing to motorcycle maintenance and from the Swedish Chef to Ronald Reagan. Usenet is simply the largest, most active, and most varied discussion forum in the world. As you might expect, there are many forums on Usenet for programmers. Here you can obtain public domain source code, ask questions about programming (and answer some), and find deliciously useful FAQs—lists of frequently asked questions and their answers.

Below is a list of newsgroups of interest to programmers. Most of their names are self-explanatory; for instance, comp.sys.mac. hypercard is a forum for discussing HyperCard on the Macintosh. Most groups are places for discussion—questions and answers, arguments, and conversations abound on those newsgroups. Others (usually with the word *source* in the name) are dedicated to source code postings. Still others (those with the word *binaries* in the name) contain only ready-to-run programs.

Here are the best and brightest Usenet groups for those interested in Macintosh programming:

- CodeWarrior discussions—comp.sys.mac.programmer.codewarrior
- programming Q&As—comp.sys.mac.programmer.help
- announcements of interest to developers—comp.sys.mac.programmer.info
- Macintosh programming discussions that don't fit in other groups—comp.sys.mac.programmer.misc
- programming tools discussions—comp.sys.mac.programmer.tools
- Newton programming talk—comp.sys.newton.programmer
- HyperCard and HyperTalk discussions—comp.sys.mac.hypercard
- discussions about PowerPC chips and computers—comp.sys.powerpc
- general discussions about programming—comp.programming
- general discussions about object-oriented programming—comp.sys.mac.oop.misc
- discussions about MacApp version 3—comp.sys.mac.oop.macapp3
- discussions about TCL—comp.sys.mac.oop.tcl
- collections of ready-to-run applications—comp.binaries.mac
- information about programming games on any platform—rec.games.programmer

You may find the comp.lang newsgroups useful, too. With these groups you can discuss any language you can imagine, from Ada to Verilog. Most aren't specific to Macintosh programming, and some are explicitly dedicated to talking about programming on other platforms. For instance, there's comp.lang.apl, comp.lang.forth.mac, comp.lang.pascal, comp.lang.scheme, and too many others to list here.

A variety of newsgroups are dedicated to source code and discussions about it. Here are some source-oriented newsgroups you might find interesting: alt.sources.mac, alt.sources, comp.sources.wanted, comp.sources.games, and comp.sources.reviewed.

FTP Site Resources

FTP stands for File Transfer Protocol. It's a tool that allows you to copy files between computers on the Internet. You can move your own files between computers, or, more commonly, use "anonymous FTP" to access huge online software libraries. Thousands of sites provide anonymous FTP service, allowing you to download everything from electronic books and magazines, to satellite pictures of the weather, to public-domain utilities and games for your personal computer. Here are a few interesting sites for Macintosh programmers:

- *mac.archive.umich.edu.* This is my favorite archive site for cool Macintosh stuff, complete with games, utilities, developer tools, virus programs, and anything else your heart could desire. Look in the directories /mac/development/source for ready-to-compile program source code and /mac/development/source/snippets for code fragments. (If this site is too crowded, you can access a copy of it at mirror.aol.com/pub/mac.)
- *sumex-aim.stanford.edu.* This Stanford University resource is one of the finest collections of Macintosh software anywhere. Sumex is home to hundreds of megabytes of Macintosh freeware, shareware, and demonstrations of commercial software. Hundreds of programs and source code snippets are available in the directories /info-mac/dev and /info-mac/dev/src. And while you're there, pick up the latest edition of the comp.sys.mac.programmer digest—containing all the good stuff posted to the Usenet newsgroup of the same name. You'll find it in the directory /infomac/ per/csmp. (If this site is too crowded, you can access a copy of it at mirror.aol.com/pub/info-mac.)
- *nic.switch.ch.* This site contains source code of all kinds, and is perhaps the most comprehensive Macintosh source code archive. It contains source for many UNIX-ported utilities as well as a lot of original Macintosh source code. Look in the directory /software/mac/src.
- *ftp.cc.umanitoba.ca.* This site is home of the *develop* magazine mailing list, and it also has lots of random stuff you can't find elsewhere. It's worth a look, so check out the directory /Mac-Develop/Source.

- *ftp.apple.com.* This is Apple's semi-official repository for system software, developer tools, source code, Macintosh technical notes, and other things that come more or less straight from Apple. Unfortunately, the materials at this site are arranged poorly, but you can unearth some true Apple treasures. You can get System 7, Tune-ups, QuickTime, and much more. (Start by perusing the /pub/dts directory.)

Mailing Lists

Even if you don't have access to the World Wide Web or FTP, chances are your Internet service includes electronic mailing lists. A mailing list is simply an e-mail address that redistributes its mail, usually on a specific topic, to other addresses. When someone sends mail to the mailing list, the message is redistributed through e-mail to the list's "subscribers." It is a way to reach a few dozen or a few thousand people who are interested in a specific topic.

Most mailing lists are available to the Internet public, so anyone interested in a topic may join that list. Here are a few lists that may be of interest to developers:

- *QuickTime programming.* This is a mailing list for discussions of QuickTime programming issues. To subscribe, send e-mail to listproc@abs.apple.com with a subject line that is blank and a text body that contains the phrase "subscribe quicktime-dev Yourfirstname Yourlastname".
- *think-c.* This is a discussion of programming with Think C. It's also a good place to look for sample source code. To subscribe, send e-mail to think-c-request@ ics.uci.edu.
- *AOCE.* This is a forum in which PowerTalk users, PowerShare administrators, and software developers share advice, problems, and so on, with each other and with Apple. To subscribe, send e-mail to aoce-list-request@umich.edu.
- *Macscripting.* This is a forum for any and all discussion about scripting on the Macintosh, whether it concerns AppleScript, Userland Frontier, or another system. To subscribe, send e-mail to listserv@dartmouth.edu.
- *apple-internet-users.* This forum is devoted to discussions on the use of Macintosh Internet client software, such as Eudora, Anarchie, and Netscape. To subscribe, send e-mail to listproc@ abs.apple.com and put "subscribe apple-internet-users Your Name" in the body of the message.
- *apple-internet-providers.* This list provides discussions on Macintosh Internet server software, including programs such as MacHTTP, FTPd, and MailShare. To subscribe, send e-mail to listproc@abs.apple.com with "subscribe apple-internet-providers Your Name" in the body of the message.

mac.archive.umich.edu is my favorite archive site for cool Macintosh stuff, complete with games, utilities, developer tools, virus programs, anything else your heart could desire.

By the way, a searchable database of Macintosh-related mailing lists is available on the World Wide Web at http://ici.proper. com/1/mac/mailing-lists.

Newsletters

A wide variety of free newsletters and magazines are available on the Internet, too. Here are a few of interest:

- *PowerPC News.* If you are interested in Power Macintosh computers, you should definitely check out the PowerPC News server at http://power.globalnews.com. It's perfect for developers and end-users who want the latest news about the PowerPC microprocessor family and all the PowerPC platforms. PowerPC News offers a free electronic magazine, published every two weeks through e-mail. To subscribe, send e-mail to add@ power.globalnews.com.
- *Edupage.* This is an excellent thrice-a-week newsletter summarizing news items on information technology. *Edupage* is a miniature *MacWEEK*—only you don't have to lie through your teeth to get a free subscription. To add your name to the *Edupage* distribution list, send e-mail to listproc@educom.edu. In the body of the message type "sub edupage Firstname Lastname".
- *TidBITS newsletter. TidBITS* is a wonderful online newsletter for Macintosh people. It's a free weekly electronic publication that reports on interesting products and events in the computer industry, with an emphasis on the world of the Macintosh. Published by Adam Engst (author of *The Internet Starter Kit for Macintosh*), it includes lively features, hot news, and roundups of all the hardware and software reviews in every other major Macintosh publication. For information on how to subscribe to *TidBITS* and where to find back issues, send e-mail to info@tidbits.com. If you're on the World Wide Web, check out the *TidBITS* home page at http://www.dartmouth.edu/ pages/TidBITS/TidBITS.html.

A Final Word

Gaining access to the Internet is well worth the effort. There isn't a library in the world that offers you as much Macintosh programming information. So next time you're up at 3 A.M., spinning your wheels on a tough programming problem, take a break and hunt for buried treasure on the Internet. ♣

Kevin Savetz (savetz@northcoast.com) is a computer journalist living in northern California. He is author of Your Internet Consultant—The FAQs of Life Online *from Sams Publishing.*

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.