The Developer Business Report *August 1996*

AppleDirections

CONTENTS

Next Month An interview with Apple Developer Relations Vice President Heidi Roizen

APPLE NEWS

Apple's New Developer Web Site Debuts

Apple Developer World Integrates Technical, Business Information

Apple Developer Relations (ADR) has unified its Internet information services for developers into a new World Wide Web site called *Apple Developer World.* The new site "went live" July 1, and we think you're going to like it.

The site offers a new design, one that's consistent with the main Apple Computer, Inc., page at **http://www.apple.com** and other new Apple Web pages, to make it easier for you to find the information you're looking for. It also provides a wealth of information—new and old—for current Apple developers, for those who are new to the Macintosh, Newton, and Pippin platforms, and for those who are considering developing for Apple's platforms.

The Apple Developer World main page, which can be found at **http://devworld.apple.com/devworld.shtml,** provides a summary of the latest Apple developer business and technical news with links to other sites for details. From the main page, you can go to the following four main areas:

- Marketing Your Products provides information to help you market Apple platform products. Perhaps we're calling this to your attention first because it contains the *Apple Directions* Web site (which is soon to be updated with the new look, as well); more important is the wealth of new business information available at this site. It includes market data—which will be updated regularly—as well as information about Apple marketing programs for developers, specifics about

please turn to page 10

STRATEGY MOSAIC

Making the Transition to OpenDoc

By Gregg Williams, Apple Directions *staff*

If you still doubt OpenDoc's place as a key piece of the Mac OS, you must not have attended last May's 1996 Worldwide Developers Conference. Here are some of the things that occurred at the WWDC:

- The conference featured 14 sessions (not counting repeats) that were related to OpenDoc. This made OpenDoc second only to Mac OS 8 in the number of WWDC sessions devoted to a single technology.
- Apple Vice President and head of AppleNet Larry Tesler declared OpenDoc (through Cyberdog, an OpenDoc-based technology) to be Apple's vehicle for integrating Internet content into Mac OS software and documents.
- Apple announced that it would use OpenDoc to deliver new technologies to users' desktops. In addition to creating viewer parts for its own technologies—QuickTime, QuickTime VR, QuickDraw 3D, sound, and 2D pictures—Apple also plans to create OpenDoc parts that will run Java applets and Netscape plug-ins.
- Netscape announced its endorsement of OpenDoc and said that, as its first project, it would create a Cyberdog browser replacement part. Spyglass also announced that it is building a World Wide Web browser part.
- Several developers demonstrated impressive OpenDoc parts. Digital Harbor (**http://www.dharbor.com**) showed an innovative word-processing part called *WAV,* and Adrénaline Software (**http://www.adrenaline.ca/**) showed a full-featured spreadsheet part and a QuickDraw 3D–based charting part that created stunning, rotatable 3D charts.

please turn to page 5

AppleDirections
Volume 4, Number 8

Apple Directions, the monthly developer newsletter of Apple Computer, Inc., communicates Apple's strategic, business, and technical directions to decision makers at development companies to help maximize their development dollar.

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/Graphic Designer
Lisa Ferdinandsen (LISAFERD)

Contributors
Peter Bickford, Dave Curbow, Alex Dosher, Elizabeth Dykstra-Erickson, Kerry Ortega, Geoff Schuller

Manager
Mark Bloomquist

Production Manager
Diane Wilcox

Prep and Print
Consolidated Publications, Inc., Sunnyvale, 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

The Business of Dreams

My ten-year-old son (and I) just went through a tough Little League baseball season.

He played on a terrific team that almost went all the way, getting as far as the semifinal round of our local Tournament of Champions before losing. He did great, never making an error in the field, doing everything he could to follow the coaches' instructions, and getting some key hits in tough situations.

The only trouble was, his coaches were so serious about winning that he didn't get too many chances to be a star. Since he sat out a lot, he had the feeling he wasn't as good as some of the other players, and he put a lot of pressure on himself when he did get the chance. All in all, it was pretty stressful: lots of tears and not as much fun as a "game" should be. (Hang in there for a minute; I'll get around to relating this to hardware and software development real soon.)

Before I go any further, for those of you who don't know about American baseball in general, and Little League in particular, it seems to me that it's all really about dreams. Unlike some sports that are, arguably, healthier for the human psyche, baseball is a tremendously individualistic sport. Heroes are made —or broken—in an instant. It's not like soccer, where the whole team is involved from beginning to end; instead, it can seem like the whole game rests on any player's shoulders at any given moment.

Kids dream about their time to be a baseball hero, and Little League is the chance for a few of them to turn those dreams to reality—or, for the majority, to find out that they need to find new dreams. (And dads of Little Leaguers sit in the stands and say "That's my kid" when their sons—or daughters—do something good, still living out the fantasy of their own Little League days.)

It seemed at times that my son was going to have some of those dreams snatched from him by the severity of his coaches' winning desires this season. Dreams are very important to him—he once accused me of "ruining his dream" of becoming a professional basketball player when I reminded him that very few people get that opportunity. My heart nearly broke those days when he'd put on his snappy uniform and run to the field full of hope that today would be the day, only to find that he was on the bench for half the game.

Whenever I talk to Macintosh and Newton developers, I find out how important dreams are for our platforms, as well. (Told you I'd get around to development issues.) Ours are the platforms where business mixes with pleasure and passion on a regular basis. Ours are the platforms where people often care more about the quality of the computing experience we give our customers than about market share. Ours are the platforms that are (still) changing the world, one person at a time. To put it in current lingo, dreams and passion are the linchpin of the business proposition.

Sharing that passion, I'm saddened when I hear Wall Street analysts and others evaluate Apple's contribution to the world strictly in terms of stock value, hard numbers, and other numeric measures of success. Yes—we have to have a business based on good numbers. But, on our side of the computer industry fence, those numbers are the result of the excitement we bring to our work, the idea—as I've heard so many of you say—that our programming, marketing, and other efforts are making a positive difference to the world.

I found out from my son that his dreams die hard. After the season was over, I asked him if he was still thinking about baseball and how bad he felt about losing his last game and not playing as much as he'd wanted to. He said, "I'm thinking a lot about baseball: I'm thinking about making a diving catch in the outfield and then jumping up to throw the runner out at home plate."

The dreams of Apple developers die hard, too. Those dreams started things off with the first Macintosh computers and software, and they're what keeps us going today.

Paul Dreyfus
Editor
dreyfusp@apple.com

For examples of the passion developers bring to the Macintosh platform, check out our new column on page 11, "Why I Prefer Macintosh."

Apple Number One in Multimedia Systems for Second Straight Year

A recent study by market researcher Dataquest names Apple Computer, Inc., the number one vendor of complete multimedia personal computer desktop systems worldwide for the second year running, with 18.8 percent of the market—4 percentage points more than runner-up Packard Bell (with 14.4 percent of the market) and double the 9.3 percent share of third-place Compaq. Including multimedia systems shipped by manufacturers of Mac OS–compatible computers, the Mac OS accounted for 19.6 percent of the worldwide 1995 multimedia market. For the purpose of the study, a *complete multimedia personal computer desktop system* was defined as a desktop personal computer that ships with, at minimum, integrated 8-bit sound capability and a CD-ROM drive.

Dataquest estimates that 20.9 million complete multimedia desktop systems shipped last year, slightly more than double the 10.4 million units shipped in 1994. For the first time, the majority of complete multimedia systems shipped to regions outside North America; Europe, Japan, Asia/Pacific, and the rest of the world accounted for 52.4 percent of worldwide shipments, up from 30.9 percent in 1994.

Apple Computer shipped 3.93 million multimedia systems last year, just over half of them—2.2 million—in North America. Europe was Apple's second largest market for multimedia systems, with 806,000 units shipped, followed by Japan (597,000 units) and Asia/Pacific (199,000). Manufacturers of Mac OS–compatible systems shipped a total of 157,000 units worldwide.

Apple's chief competitor was Packard Bell, which outpaced Apple in the U.S. market with a total of 2.4 million units shipped but placed second worldwide with 3 million units sold. In Japan, Apple faced competition from NEC, which shipped 1.29 million units, and Fujitsu, with 571,000 units sold. Worldwide, Compaq had the third largest share, selling 1.9 million units, followed by IBM (7.4 percent, 1.55 million units), and NEC (7.0 percent, 1.47 million units).

Implications/Opinions: It's clear that customer demand for multimedia capabilities is exploding. With its leadership in interactive media technologies such as QuickTime, which has also helped make Macintosh computers the choice of the majority of cross-platform multimedia content, Apple is in a strong position to take advantage of that demand. Apple CEO Gil Amelio identifies the demand for media-rich digital content as one of the industry "megatrends," and he's made it Apple corporate strategy to stay ahead of that and other trends. Says Satjiv Chahil, senior vice president of corporate marketing for Apple, "Apple will continue to innovate and capture the hearts and minds of our customers and multimedia developers alike; in the year ahead, watch for initiatives from us that directly and positively impact multimedia developers, consumers, and retailers." Apple is committed to staying on the cutting edge of interactive media; an example of that is its Cyberdog technology, which makes it easy to build media-rich documents with Internet content. It's important that you stay ahead of the multimedia curve, too, if you're going to provide customers with truly exciting products.

September *Apple Directions* Online

September's *Apple Directions* will be available by August 15 at the following locations:

- AppleLink path—Developer Support:Developer Services:Periodicals:Apple Directions
- Internet—**http://dev.info.apple.com/appledirections/adtoc.html**

More About DVD

Philips and Toshiba announced earlier in the year that the first DVD-ROM drives will hit the market by the end of this year. Apple has also committed to delivering the first DVD-ROM–equipped Macintosh systems by late 1996 or early 1997, and says it will aggressively push DVD into the entire desktop product line in 1997.

As we said in the March 1996 installment of this column, DVD—the name isn't an abbreviation for anything—is the emerging standard for CD-based content and software. DVD discs are 5.25-inch optical storage disks with up to 25 times more storage capacity and 9 times the speed of today's CDs. Its increased capacity allows DVD to store high-quality video as well as higher-than-CD-quality digital audio. DVD discs will come in DVD-Audio or DVD-Video, audio/video storage media for use with a DVD player connected to a stereo system or television, and DVD-ROM, a computer data storage medium for use with a DVD-ROM drive. DVD represents a potential breakthrough in the convergence of home electronics and computing, since a variety of firms in the entertainment, home electronics, and computer industries are working together to set standards for how the discs utilize the data stored on them, including audio, video, movie, and application software. Many analysts expect DVD-ROM drives to rapidly take the place of CD drives; for a relatively small increase in cost, they promise far greater flexibility, since they play both today's CDs and tomorrow's DVD-based discs.

Implications/Opinions: In March, we urged you to read up on the new technology. Now, if you deliver software or content on CD, we think it's increasingly important for you to learn everything you can about DVD. As the previous item pointed out, Apple will be pushing a variety of technologies to expand its lead in multimedia; early support for DVD will give Apple platforms an immediate technological boost over the competition. Apple expects to focus its initial DVD marketing efforts on the market for high-end multimedia authoring systems as well as in the education and home markets, where multimedia capabilities are very important to customers. If you're among the first to adopt DVD and ship discs using the new standard, you'll be able to deliver unheard-of quantity—and quality—of information to your customers. You'll also be in the spotlight as you break ground with the new standard. For more information about DVD, we recommend that you visit the DVD home page at **http://www1.usa1.com/~philk/dvd/index.html**, which also contains links to information about DVD at the Toshiba, Philips, and Sony Web sites. (This is not an official Apple site, so our mentioning it here cannot be considered an endorsement.)

Macintosh Computer Again Leads Industry in Brand Loyalty

Computer Intelligence, a market research firm based in La Jolla, California, says in its annual Consumer Technology Index (CTI) study that the Macintosh computer led the U.S. PC industry in repurchase loyalty again in 1995, as it did in 1994. Following Apple were Dell, Hewlett-Packard, Acer, and Gateway 2000, ranking second through fifth. IBM, Compaq, AST Research, Packard Bell, and NEC completed the list of top ten brands, in that order. According to the CTI results, about seven in eight Macintosh users who purchased a computer in 1995 purchased another Macintosh system. (See "U.S. Repurchase Rates for Major Personal Computer Brands, 1995" below.)

The study looked at customers in two categories: home and business. Those who purchased computers for personal use generally showed lower brand loyalty than individuals who work on employer-provided PCs. For example, with Dell, Hewlett-Packard, and Acer, repurchase loyalty in the workplace segment was more than twice as high as in the home segment, where customers tend to be far more conscious of how much they spend. Even in the price-conscious home market, Macintosh users proved highly loyal. In fact, Apple's results for the two segments are the most nearly equal of all the major manufacturers, with a repurchase rate of 81 percent among home customers and 91 percent in the workplace.

Implications/Opinions: We'll step aside and let David Tremblay, CI's senior industry analyst for PCs, provide the analysis. In issuing the report, he said, "In spite of the highly publicized troubles Apple suffered in late 1995, Macintosh users remained the most loyal users of all PC brands, with repurchase rates in 1995 nearly identical to what we measured for 1994." He added, "Macintosh users' loyalty is a key to Apple's future. Sales to existing Macintosh users provide a revenue base to support the company while Apple develops new products in new sectors of the industry that can provide for its future growth."

U.S. Repurchase Rates for Major Personal Computer Brands, 1995

Brand	Repurchase rate
Macintosh	87%
Dell	74%
Hewlett-Packard	72%
Acer	68%
Gateway 2000	66%

Source: CI Consumer Technology Index, 1996

Katz Media to Launch Pippin in Europe

Speaking of those new sectors, Katz Media AS recently announced that it will ship computing products based on Apple's Pippin technology for the European market later this year. The products, which Katz is calling *information appliances,* are intended to allow easy, affordable access to any kind of digital information through a standard television set. Pippin is Apple's specification for easy-to-use, inexpensive multimedia Power Macintosh computing devices that can be used with a television set or monitor to provide access to CD-ROM–based data, the Internet, or both. Katz Media intends to ship a CD-based console along with a broad range of game, educational, reference, personal productivity, and other CD titles in selected European markets by the fourth calendar quarter of 1996. The company says it will bundle its Pippin device with a keyboard, a modem, and a hand-held controller. Additional peripherals such as Zip drives, floppy drives, hard disks, and other devices, will be available. Katz Media will also develop relationships with Internet service providers to deliver Internet access right out of the box. Further details about Katz Media can be found on the Web at **http://www.Katz.no**.

Implications/Opinions: Katz's announcement means that by the end of this year, Pippin devices will be available in the major geographic markets. Bandai Corporation recently released its Pippin device, called the @Mark, in Japan, and announced it will release a U.S. version—the @World—later this year. With price tags of under U.S. $1,000, the Pippin devices are aimed at the 93 percent of the world's households that don't yet own computers. Many industry experts anticipate huge demand for so-called information appliances. In a recent installment in his well-read Personal Technology column, *Wall Street Journal* writer Walter S. Mossberg said, "Such devices could attract the majority of homes and small businesses that lack a PC. They would also appeal to current PC owners as a second or third machine." Taken together, these points suggest that there will be a rapidly expanding, potentially vast new customer base for your CD-based Mac OS titles, since thousands of titles already running on Mac OS–compatible computers can easily be modified to run on Pippin-based products. Further, any Pippin title can be played on any suitably configured Mac OS–compatible computer. ♣

Making the Transition

continued from page 1

WorldSoft (**http://www.worldsoft.com/**) demonstrated their multilingual WorldWrite processor as a container application. In addition, Claris Corporation showed a version of ClarisWorks that was a container application (that is, an application whose documents can contain OpenDoc parts).

Apple's message at the WWDC was that it will use OpenDoc to keep the Mac OS a vital, innovative platform—and I think that most developers who attended the WWDC heard and understood that message. When I asked developers what they had gotten out of the conference, many of them said things like, "Well, I guess it's obvious now that OpenDoc is pretty important."

Yes, it is obvious that OpenDoc is pretty important, but that's not enough. Apple needs your help to make its vision for the future happen. In particular, Apple needs you to make your applications into container applications and to create new OpenDoc parts. If you don't make your software OpenDoc-aware, your customers won't be able to use Cyberdog to put Java applets or Web site links in their documents, and they won't be able to enrich their OpenDoc-aware documents with innovative parts.

The only question is, how do you get from here to there?

This article is about how Apple thinks you should make the transition to OpenDoc. The route you take depends on what kind of code you have now, and what you want to do in the future. There are three possibilities:

- You may want to turn your application into a container application. This makes the most sense if your application's documents would benefit from having other kinds of content (implemented as OpenDoc parts) embedded in them. Remember that a container application can also take advantage of the Cyberdog parts, thus allowing customers to embed Internet content into their documents. In addition, you can use Cyberdog to build Internet access into the application itself.
- You may also want to turn your application into a container part—that is, a part that contains other parts. By doing so, you retain all the advantages of being a container application—plus, customers can embed your content into their own OpenDoc documents.
- If you're designing a new product now, you may want to implement it as OpenDoc parts instead of an application. In most cases, you should probably do so using the OpenDoc Development Framework (ODF), which makes it easier to create high-quality parts that can easily be moved, if desired, to other computer platforms.

Apple will use OpenDoc to keep the Mac OS a vital, innovative platform.

The rest of this article will talk about various routes to implementing container applications and OpenDoc parts.

Container Applications

Making your application into a container application is the fastest way to enable your software to participate in the benefits that OpenDoc offers. Your software is still an application, but users will be able to embed any OpenDoc part into the documents your application creates. One added plus—since a container application is still an application, it will continue to run correctly on computers that do not have OpenDoc installed.

Making your application into a container application has two immediate and compelling benefits: instant support of Internet services, and immediate implementation of key data types in your application.

Your newly converted container application will gain access to the Internet through Cyberdog parts. Your documents can contain Web browsers or buttons that connect users to various Internet services. Even if you don't think it makes sense to make your application into a container application, the ability to embed a "cyberbutton" in your application is a strong reason to do so. Using the tools described later in this article, you can pretty easily add to your About This Application box a button that will take users to your Web site. There, you can offer anything you want: company information, online help, software updates, or anything else that your users will value. Furthermore, you can then put a "Supports Internet" sticker on your application and resell it to your installed base as an upgrade. (As many of you know, you got new and upgrade sales just from having an "Accelerated for Power Mac" sticker on essentially the same product. Apple expects the same to be true for developers who provide even modest levels of Internet connectivity in their products.)

Does your application currently support QuickTime, QuickTime VR, QuickDraw 3D, sound playback, and images in PICT, GIF, JPEG, and TIFF formats? If the answer to any of these is "no," you've got another reason to make your application into a container application. Later this year, Apple will be bundling a series of free "Kickstart" parts that includes viewers for all the data formats mentioned above. (You will also be able to license these parts for inclusion with your application.) If you make your application into a container application, your customers can automatically drag these data types into their documents. And it will take you only a little work to add menu items that integrate these data types into your container application; all the menu item has to do is embed the appropriate part inside your document.

Making the transition to a container application is also important because it's the first step in moving to OpenDoc-based software. When you add new features to your container application, Apple hopes you will do it by adding OpenDoc parts. There are several advantages to doing so. It's easier and faster for you to build, test, and maintain a part than to add functionality to an application. In addition, your customer can use your part in other OpenDoc-aware software, and you may be interested in licensing or bundling your part to provide a new source of revenue. Over time, an increasingly larger fraction of your solution will be implemented as OpenDoc parts. At some point, it will make sense for you to convert the remaining application-based functionality to parts, thus making your software solution (we can no longer call it an *application*) into a set of OpenDoc parts.

Son of CALib

Some time ago, Apple delivered CALib, a library of source code that was supposed to help you convert your application into a container application. Unfortunately, when CALib was delivered on the OpenDoc DR4 CD, CALib was at a pre-alpha stage, and it was not yet ready for use by most developers.

Fortunately, Claris Corporation and 6prime Corporation have stepped in with a solution. Engineers from these two companies are in the process of correcting many of the difficulties in using CALib. The result of this effort is the Claris Container Library (CCL). This is the vehicle that Claris is using for container support in the upcoming version of ClarisWorks, promised by the end of this year. If you saw the ClarisWorks demo at the WWDC, you saw an alpha version of this library running. A pre-alpha version of CCL is on the OpenDoc DR5 CD.

This new library is available in two forms. The Claris Container Library, available from Claris on the OpenDoc Developer Release series of CDs, is unsupported and is intended for developers on a tight budget or interested in exploring the technology on their own. 6prime Corporation, working in conjunction with Claris, is making significant refinements to the Claris Container Library, resulting in a supported, commercial product called OpenContent.

CCL/OpenContent is a library of routines that stands between your application and OpenDoc, allowing you to implement container support without calling OpenDoc directly. This procedural API (application programming interface) offers everything you need to add OpenDoc container support to your application. CCL/OpenContent works for all kinds of existing applications: Power Macintosh applications, traditional 680x0 applications, and 680x0 applications that use the 680x0 Code Fragment Manager.

Of course, the most important question you want to have answered is, "How much work is it going to be to convert my application to a container application?" The answer is, "It depends—but you definitely need to do it."

If you are not changing the document format, converting to a container application and using Cyberdog to implement Internet access through "cyberbuttons" is very easy. If you are changing your application's document format so that it supports embedded parts, you will have more work to do.

The ease of conversion depends on the internal design of your application. Any OpenDoc part that gets dragged into your application's document is essentially a chunk of foreign data (sometimes called a "figure type") that your application must know how to deal with—including reading it into your program from a document file, writing it out, displaying it, and dealing with user events focused on it.

The only guideline I can offer is this: If your application already deals with different figure types (pictures or QuickTime movies, for example), you will have an easier time converting to a container application than you would with an application that understands only a single figure type. Essentially, with an application that supports only a single figure type, you will have to redesign your application to add the support for multiple figure types. If your application already supports at least two figure types, you've already done that work.

6prime and OpenContent

Claris Corporation is making the Claris Container Library available to developers on an "as is" basis. However, Claris cannot provide support of the Claris Container Library to developers.

This is where 6prime Corporation (**http://www.6prime.com**), which bills itself as "a Macintosh software development house specializing in OpenDoc development and porting to the Macintosh," comes in. 6prime is enhancing the Claris Container Library code and selling it as the supported, commercial product named OpenContent. (Claris and 6prime have a technology-sharing license agreement where improvements to the library from either company are made available to the other. Either company can then pass these improvements on to developers, and both are in the process of doing so.)

However, the Claris/6prime agreement governs only the CCL layer of CCL/OpenContent. 6prime has added additional features:

- OpenContent delivers a feature called "OpenDoc on demand and if available." Loading the OpenDoc library adds about 1 MB to a container application's memory footprint, and Cyberdog adds an additional 1 MB. In its simplest implementation, a container application would load OpenDoc when it launches. However, this would waste memory if the document opened contained no OpenDoc parts. OpenContent delays the loading of OpenDoc until it is needed, thus reducing your container application's footprint in many situations. OpenContent also ensures the correct operation of your container application when OpenDoc is not available.
- OpenContent has its own API on top of the Claris Container Library API. The OpenContent API includes functions that further simplify the process of converting your application into a container application.
- In the DR3 release of OpenContent (the current release is DR2), OpenContent will make it possible for you to resolve a URL (universal resource locator, or Web site address) with a single call. In essence, this call uses Cyberdog to bring up a Web browser and display the Web page named in the call.

Container Application and Non-OpenDoc Computers

Some developers are concerned that once the user embeds an OpenDoc part into the document, non-OpenDoc users will not be able to open the document. However, this is not the case. ClarisWorks documents with embedded OpenDoc parts, for example, can be opened on machines with or without OpenDoc. This behavior is managed at the application level and involves the application's document format. Because of this, there's no way that either CCL or OpenContent can help with this situation. However, there are recipes that show you how to solve this problem. You should take a look at the *MacTech Magazine* article called "OpenDoc: Contain Yourself"; see the Resources box (page 9) for details.

Container Applications vs. New Parts

In some cases, you may find it easier to rewrite your application as a container part than to change it to a container application. Even if it's somewhat harder to make the transition to being a part, doing so is probably worth the effort. Parts are, ultimately, more valuable than container applications; they are worth more to your customers, and for a longer period of time.

If you convert your application to a container part, it will behave exactly as a container application would—that is, it will perform the functions it always used to, and customers will be able to drag other parts into it. However, the advantage of your application becoming a container part is that your customers will be able to drag *it* into OpenDoc documents along with other parts. Customers will find more uses for your application-that-is-now-a-part, and they will value it because of its status as a full OpenDoc part.

Container Applications: What You Can Do Now

Now's the time to start working on converting your application to a container application. The first released version of the Claris Container Library is on the OpenDoc DR5 CD; use the pathname OpenDoc DR5:OpenDoc Parts:Third Party Parts:Claris Container Lib. The contents of the OpenDoc DR5 CD are also on the Web (**http://www.opendoc.apple.com/**). Future OpenDoc DR CDs will contain the most current version of the Claris Container Library.

You should contact 6prime for more information about their OpenContent product (phone 408-252-9828, e-mail opencontent@6prime.com or info@6prime.com). It is constantly being improved with each developer release, and 6prime is committed to providing container support at a reasonable price.

Shaan Pruden, of Apple Evangelism, is dedicated to supporting component software (which includes OpenDoc, Java, and Netscape plug-ins). If you need help in getting your OpenDoc-based product to market, you can contact her at pruden@apple.com.

Container Applications: Apple's Future Plans

Apple is currently working on a second-generation container-support library that will minimize the amount of work it will take to convert an application to a container application; Apple expects to deliver this library sometime in 1997. However, Apple recommends that you work with the tools that are available today—the Claris Container Library and OpenContent—to implement container support.

OpenDoc Parts

Apple hopes that you will join the growing ranks of developers who are creating OpenDoc parts. Apple expects OpenDoc to take off in the next year, and developers who have OpenDoc parts ready to sell will be well positioned to take advantage of OpenDoc's increasing popularity. (For a list of reasons why OpenDoc is important to you, see the OpenDoc section of "Why Mac OS 8 Is Important, Part 2," in the July 1996 issue of *Apple Directions.* The OpenDoc section starts on page 9. On the Web, you can find this article at **http://dev.info.apple.com/appledirections/july96/stratmos.html**.)

If you are interested in creating parts, Apple recommends that, in most cases, you use the OpenDoc Development Framework (ODF) to do so.

Advantages of Using ODF

There are two main advantages to using ODF: it's a framework, and it's cross-platform.

Apple recommends that, in most cases, you use the OpenDoc Development Framework (ODF) to create parts.

What Is a Framework? A framework is a body of source code that implements an object-oriented hierarchy of software classes that are useful in building a certain kind of software. A framework also includes source code that uses these classes to implement a "skeleton" that the developer enhances to create a given type of software. ODF includes source code that implements a skeleton part, leaving you to add the code that implements the functions unique to your part.

What makes ODF—or any framework—important is the amount and quality of source code in the framework that, consequently, you don't have to write. Every OpenDoc part contains a body of functions and behaviors that need to be implemented the same way in every part, and the ODF source code implements many of these functions and behaviors for you. For example, ODF implements many parts of the human interface—how menus drop down and how a window gets activated, for example. In addition, ODF vastly simplifies the implementation of key OpenDoc features that would otherwise require significant work to implement directly. Among the features implemented by ODF are linking, embedding, multiple undo, scripting, data interchange, and storage.

The ODF framework source code delivers several benefits to you:

- It takes you less time and money to write, debug, and maintain your ODF part, and you spend more of your time writing the code that is unique to your part.
- Numerous functions in an ODF part have been written in a standard way that has been refined by Apple engineers and coordinated with the behavior of other parts of the framework. Using ODF results in a part that is more robust and stable.
- The ODF part implements the Mac OS human interface in a standard, "approved" way.
- For an OpenDoc part to be well designed, the programmer must have experience with OpenDoc and know some fairly obscure things about how to code a part well. If you use ODF, you automatically gain the expertise of experienced OpenDoc programmers, and your ODF part will pass the CI Labs part-validation process (which confirms that a part operates properly) with fewer problems.

A Cross-Platform Solution. Another main advantage of ODF is that it's designed to be cross-platform. ODF currently supports the Mac OS, and it will support Microsoft Windows approximately one quarter after IBM delivers OpenDoc for Windows 1.0. (IBM has publicly stated that it will deliver OpenDoc for Windows 1.0 in the third calendar quarter of 1996.) Finally, Apple is investigating ODF support for other platforms where OpenDoc will be available, including IBM's AIX (its implementation of UNIX®) and OS/2.

ODF interacts with various computer subsystems—the file and graphics systems, for example—in a platform-independent way. Taking the example of graphics, this means that when you write an ODF part, you call ODF drawing routines to draw into an abstract graphics space. Then when you compile your part for a certain platform, the code in the framework that implements the ODF drawing routines makes the appropriate calls to the platform's graphics primitives. (In contrast, OpenDoc does not address several important subsystems—indeed, its platform-neutral design makes it impossible to do so. So if you're writing a part straight to the OpenDoc API, you need to do such coding—your drawing routines, for example—yourself, and you have to rewrite that code for each additional platform you support.)

Because the ODF programming model is platform independent, you will need to change relatively small portions of your ODF part's source code to create a new version of your part that works on another platform. This means that, by using ODF, you are potentially developing for a much larger customer base than you would be otherwise.

When to Create Parts Using ODF

Apple believes that ODF is, in general, the best way to create OpenDoc parts. It's faster for you, it increases the stability and robustness of your part, and it allows you to enlarge the market for your part by going cross-platform.

However, no development situation is without trade-offs, and ODF is no exception. Frameworks always carry an overhead with them (in the case of ODF, a memory-footprint overhead), and Apple engineers are working on ODF to decrease that overhead.

However, Apple engineers have already found three ways to possibly reduce the memory footprint of ODF parts. ODF is based on an architecture that has three layers: from top to bottom, the framework, OS, and foundation layers. The layers are divided into 23 separate subsystems, and ODF is designed so that you don't have to include subsystems that you don't use; in some cases, this modularity may allow you decrease the size of your part.

The second, most significant footprint reduction occurs automatically if your customer has multiple ODF parts (either yours or from multiple vendors) running on a Mac OS–compatible computer. To decrease ODF's overall memory footprint, Apple engineers implemented most of the OS and foundation layers of ODF as a shared library named *ODFLibrary.* (ODFLibrary must be present in any computer that is using ODF parts.) The presence of ODFLibrary means that, as the number of ODF parts in use increases, the overhead per ODF part goes down. (At some point, the memory footprint of *N* ODF parts will become less than that of the same parts written straight to OpenDoc.) Because of ODFLibrary, you should not worry about ODF's memory-footprint overhead if you are creating a suite of parts meant to be used together.

This brings me to the third way to possibly reduce ODF's memory-footprint overhead. If you are creating multiple parts meant to be used together, you can reduce their overall footprint by binding them into one shared library. For more details, see the document *Merged Part Suites* on the OpenDoc DR5 CD, pathname—OpenDoc DR5:OpenDoc Development Framework:Documentation:Development Notes.

Before you start worrying about whether the overhead of an ODF part is enough to deter you from using ODF, you should factor in the complexity of your part. Apple engineers estimate that an ODF part has an overhead of 100K to 150K, with the overhead decreasing rapidly with the complexity of the part. (The ODF code includes functions you would have to add yourself in a more complex part.) This means that the overhead is most visible in a simple part; the more complex the part, the less significant ODF's overhead turns out to be.

Using the OpenDoc Development Framework results in a part that is more robust and stable.

When to Create Parts Using the OpenDoc API

From the previous section, you can pretty much infer when to create parts by directly coding to the OpenDoc API, but I'll spell it out explicitly:

- when you're building a simple part (usually a viewer) that doesn't support embedding, scripting, or linking
- when it's a priority to minimize the part's memory footprint

If you decide to build a simple OpenDoc part without using ODF, you should use a tool called *PartMaker,* which automatically creates a lot of OpenDoc source code on which you can base your simple part. You can find it on the OpenDoc DR4 CD. (The version of PartMaker that is on the OpenDoc DR5 CD uses ODF and so would not be helpful here.) You can get a copy of a non-ODF version called PartMaker Pro on the Web (**http://www.opendoc.apple.com/dr-live/OpenDocDevelopment/byName.html**).

ODF: What You Can Do Now

The first thing you should do is read the *ODF Developer's Guide,* especially Chapter 1, which gives overviews of the ODF architecture and ODF parts. On the OpenDoc DR5 CD, you can find this document at OpenDoc DR5: OpenDoc Development Framework:Documentation:ODF Dev Guide (Acrobat). This document is also available on the OpenDoc DR LIVE! Web site (**http://www.opendoc.apple.com/dr-live/index.html**).

The next thing you should do is create a simple part using OpenDoc. ODF is based on OpenDoc, and you will learn ODF easier if you've had some experience with OpenDoc.

Your next step is to start using ODF 1.0 to create parts. The most important piece, ODFLibrary 1.0, is of "version 1.0" quality, and the framework (pieces of which are at the alpha and beta stages) is good enough to use for shipping commercial products. See the document "ODF Release 1 Development State" (supplied as part of ODF 1.0) for details. Also, keep in mind that ODF 1.0 includes the entire source code of the framework. This means that you can change or add to the framework to solve whatever problems you might have.

ODF 1.0 is included on the OpenDoc DR5 CD. If you don't have that CD (it's been sent out to developers through various channels), you can access the latest OpenDoc Developer Release from the OpenDoc Web page (**http://www.opendoc.apple.com/**), or you can get the latest OpenDoc Developer Release CD by sending an e-mail request to opendoc@apple.com.

To obtain support and resolve problems regarding ODF development, join the odf-interest mailing list, where third-party ODF developers and Apple ODF engineers offer help and respond to questions. To join this mailing list, send e-mail to listproc@cilabs.org; in the body of the message, type "join odf-interest <your full name>".

ODF: Apple's Future Plans

Apple will continue improving the framework source code in future releases, which will occur several times a year. Also, Apple will supply a cross-platform version of ODF that will allow you to create Mac OS and Windows parts as soon as possible after IBM releases OpenDoc for Windows 1.0.

Future versions of ODFLibrary will include bug fixes and enhancements on an as-needed basis. However, Apple will modify ODFLibrary in a way that will ensure release-to-release binary compatibility for existing ODF parts—that is, ODF parts compiled to run with, say, ODFLibrary 1.0 will work correctly with later versions of ODFLibrary.

MacApp

A good number of you—especially those of you who write applications for vertical-market or in-house use—use the MacApp framework. If so, you have two routes to providing container support: either use Claris Container Library or OpenContent now, or wait until Apple integrates container support into MacApp sometime in 1997.

Apple's intent is to add a container-support architecture to the MacApp architecture, such that Apple can add a different container-support library (for example, the second-generation container-support library mentioned earlier) without forcing you to rewrite your MacApp source code. Apple has already started with MacApp Release 10 by adding rudimentary support for the CALib container-support library. (MacApp Release 10 is currently available and can be found on the following software development disks: E.T.O. #20, CodeWarrior #9, and Symantec #5.)

You can see a sample application that delivers container support by looking at the sample project called SimpleContainer. (SimpleContainer works only for PowerPC Mac OS–compatible computers and the CodeWarrior compiler. Sometime after MacApp Release 11 comes out on E.T.O. #21, Apple will post an update to SimpleContainer that will allow it to work with the OpenDoc DR5 release and MacApp Release 11. The update will be posted on the MacApp Web site (**http://www.devtools.apple.com/macapp/**).

MacApp: What You Can Do Now

According to Geoff Clapp, one of the MacApp engineers, you can get a feeling for how Apple will implement container support within MacApp by reading the MacApp Release 10 Release Notes (on the E.T.O. #20 CD, pathname E.T.O. #20:Frameworks:MacApp Release 10:Release Notes) and studying the SimpleContainer sample code.

After you've studied SimpleContainer, you should start thinking about how embedded parts will affect your MacApp application. Remember that any embedded part in your document is essentially a "black box" embedded within your document. What does that mean to your application's structure and behavior? How will you deal with reading and writing the data from embedded parts into and out of your documents? Also, how will you handle opening a document containing embedded parts on a computer that doesn't have OpenDoc installed?

Also, if you want to use the Claris Container Library to add container support to a MacApp application, you should participate in the comp.sys.mac.oop.macapp3 newsgroup on the Internet. Doing so will allow you to air important issues and receive feedback from the developer community and the MacApp engineering team.

MacApp: Apple's Future Plans

Apple will continue to refine the container support of MacApp in future releases, using more robust container-application libraries as they become available. Expect to see a version of MacApp with container support sufficient for shipping commercial applications sometime in 1997.

Resources

- Soldan, Eric. "OpenDoc: Contain Yourself." *MacTech Magazine,* January 1996, 71. This article shows how to use CALib to add container support to an application called *DTS.Draw.*
- Çelik, Tantek and David Curbow. "Rethinking the Interface." *MacTech Magazine,* May 1996, 71. Discusses the changes you need to make (mostly changes to your application's human interface) when you convert it into a container application.
- Shaan Pruden, pruden@apple.com, is the Apple evangelist for OpenDoc, Java, Netscape plug-ins, and all other component software.
- Check out the various Web sites (highlighted in color) mentioned in the body of this article.

Conclusions

At this year's WWDC, Apple made it clear that OpenDoc is the key to accessing many important future technologies—not the least of which are Internet connectivity and access to new Apple and third-party technologies (such as Java applets and Netscape plug-ins).

In addition, OpenDoc will become more visible by the end of 1996. Apple will introduce OpenDoc to customers at Macworld Boston in August. Later this year, OpenDoc will ship on suitably configured Macintosh models. Apple will also be shipping a basic set of parts (the Kickstart parts mentioned earlier) that will make OpenDoc immediately useful. The Kickstart parts will stimulate demand for your OpenDoc parts. You can also license them to ship with your parts, thus cutting down on the amount of work you have to do to ship a completed product. (For details on the Kickstart parts, see the OpenDoc section of "Why Mac OS 8 Is Important, Part 2," in the July 1996 issue of *Apple Directions*.)

Let's not forget that OpenDoc itself is a good reason for its own adoption. Component software (in the form of OpenDoc parts) is an exciting new paradigm for delivering software to customers. It allows you to create and revise software more quickly. It allows you to concentrate on your strengths and not be forced to deliver functions that other parts already deliver. It offers new business opportunities and helps level the playing field, so that small companies have a better chance at creating a sustainable business.

But as I said at the beginning of this article, Apple needs your help. The Kickstart parts will make OpenDoc useful to customers, and Apple is planning on using OpenDoc in future products and operating-system enhancements. If you don't create parts or convert your applications into container applications, OpenDoc cannot live up to its potential. However, if you do help, both you, Apple, and the Mac OS platform will benefit.

As this article shows, you can begin to create parts or convert your applications to container applications today. Apple strongly recommends that you do so: It's time to take the next step forward. ♣

New Developer Web Site

continued from page 1

opportunities for you to comarket with Apple (as those opportunities become available), as well as the It Shipped and Third-Party Product databases.

• Developing Your Products is for technical folks. It provides you with access to the plethora of online technical support materials from Apple, including Technotes, technical Q&As, Sample Code, Developer University class information (including a variety of self-paced online classes), software development kits, tools, and a variety of other technical resources.

• Considering Apple Platforms is aimed at developers who are thinking about developing for Macintosh, Newton, or Pippin systems for the first time. It includes Apple's recently revised *Business Proposition for Developers,* a separate section with resources to help new developers get started with Apple, and a technical brief about porting applications from the Windows operating system to the Mac OS.

• Ordering Development Products gives you direct access to the Apple Developer Catalog (ADC) Online, Apple's source of tools, books, and technical resources for development on all Apple platforms. Go to this area to order Apple and third-party development products, information, and services.

The Developer World main page also contains a list of links to Apple's technology Web sites (such as the Mac OS 8, OpenDoc, Cyberdog, and Open Transport Web pages) and a Cool Areas list that can take you directly to specific areas within Developer World. The Find It button takes you to a page that provides four different ways to find what you're looking for: an alphabetical index, a site table of contents, a graphical site map, and a free-form search engine.

We'll tell you more about Apple Developer World in forthcoming issues, and we'll do our best to let you know about relevant new postings to the site. For now, we suggest you log on to **http://devworld.apple.com/devworld.shtml** to experience Developer World firsthand—and please let the Developer World team know what you think by sending your feedback to dpfeedback@applelink.apple.com.

■ ■ ■

System 7.5.3 Offers Latest Macintosh Upgrades, Internet Connection

The latest Mac OS upgrade, Macintosh System 7.5.3, provides enhancements to overall system stability and performance improvements for Macintosh and Mac OS–compatible computers.

The new version of the Mac OS, which shipped to U.S. customers in early July and will be available in other markets by late September, includes the Apple Internet Connection Kit, Apple's software bundle that provides easy access to the Internet for Macintosh customers. System 7.5.3 is the first release of the Mac OS to include OpenDoc; it also updates major system features and technologies for network communications and multimedia. In addition, it offers all the system reliability and performance improvements included in previous nonretail releases, including System 7.5 Update 2.0 and System 7.5.3 Revision 2 as well as the LaserWriter 8.3.4 driver.

Key features of System 7.5.3 include convenient one-button installation for all supported systems; a universal System Folder that can start up any Macintosh system, from the Macintosh Plus computer to currently shipping systems; improved installation assistance; and reduced software installation time. These features enable system administrators and technical support staff to upgrade and support large organizations with a broad range of Macintosh computers.

System 7.5 Version 7.5.3 delivers performance improvements with several new PowerPC native components. It also improves stability—for example, by eliminating some causes of type 11 errors on PowerPC processor–based Macintosh computers. The new system also provides several feature enhancements; for example, comments are retained in the Info window after the desktop is rebuilt. The update also includes detailed documentation on the contents of the System Folder, the Control Panels folder, and the Extensions folders.

The release also offers new capabilities to Apple's networked users with the integration of Open Transport 1.1, the latest standards-based networking and communications system for the Mac OS, which improves performance for many Internet related tasks. Other key technologies updated in System 7.5 Version 7.5.3 include multimedia creation and viewing with QuickTime, advanced graphics with QuickDraw GX, interactive help with Apple Guide, easy three-dimensional graphics with QuickDraw 3D, text-to-speech conversion with PlainTalk, and enhanced printing capabilities with desktop printing.

Included with Mac OS for the first time is OpenDoc, the multiplatform component architecture developed jointly with IBM. This will give System 7.5.3 customers access to your OpenDoc components and containers. Users will be able to customize their workspace by simply dragging and dropping components—including connections to the Internet—to their desktops.

Included in System 7.5 Version 7.5.3 is the Apple Internet Connection Kit, an integrated collection of Apple and third-party software that allows a Macintosh computer user to connect directly to the Internet. It includes more than a dozen software applications and tools, including Netscape Navigator™ for the Macintosh and Claris Emailer Lite. The release also includes access to America Online, which also provides easy and affordable access to services on the Internet. The Internet products bundled with System 7.5.3 are different outside the United States.

The estimated retail price of System 7.5 Version 7.5.3 is U.S. $99, which includes the Apple Internet Connection Kit, MacLink Plus file translators, and access to America Online. Current System 7.5 customers can upgrade to the new version for U.S. $49 by calling 800-293-6617 ext. 1, and providing proof of ownership. The upgrade consists of System 7.5 Version 7.5.3 and the Apple Internet Connection Kit. Localized versions of this system release will be available to customers in many countries outside the United States by late September.

■ ■ ■

Free Cyberdog, Mac OS 8 Online Courses from Developer University

Developer University (DU) recently announced two new online training tutorials, Get Started Programming with Cyberdog and What Is Mac OS 8? The Cyberdog tutorial illustrates how to convert a basic OpenDoc part into a fully functioning Cyberdog component. The tutorial contains links and references to online Cyberdog documentation. It explains the what, why, and how of each step toward getting started with Cyberdog programming. The Mac OS 8 tutorial is a short, nontechnical overview of Apple's next-generation operating system. Check out these online courses and others at the DU web site (**http://dev.info.apple.com/du.html**). For a complete list of courses from DU, see page 30 of this issue.

■ ■ ■

Newton Internet Enabler Now Available

Apple Computer recently shipped the Newton Internet Enabler (NIE) 1.0 for Newton 2.0, a suite of connectivity tools that allows Internet-based application development for Newton hand-held computers, including the Apple MessagePad.

NIE will enable Newton developers to create Internet-based solutions such as Web, mail, news, Telnet, and FTP applications. NIE provides support for standard Internet and communications protocols, such as HTTP, HTML, SMTP, POP3, IMAP4, PPP, SLIP, and UDP. The suite also includes a dialer with a script engine and a domain name resolver (DNR). Using NIE resources, software applications allow Newton users to interact with TCP/IP networks through public Internet Service Providers (ISPs) and private corporate intranets.

"NIE is a key component of our Internet strategy for the Newton platform," said Jim Groff, senior vice president of Apple's information appliance division. "It will drive an exciting range of innovative Internet-based solutions for Newton, developed specifically for the mobile customer who requires seam-

Prefer Macintosh... Why I Prefer Macintosh... Why I Prefer Mac

Why I Prefer Macintosh

To get this new column started, in the July 1996 issue of *Apple Directions,* Editor Paul Dreyfus asked readers to send in their answers to the following question:

Why do you prefer to develop on/for the Macintosh computer instead of a Windows-based PC?

Following are the first responses.

• • •

From Keith Schindler, Project Leader (Macintosh)
Pixera Corporation
keith@Pixera.com

As the literature for our product, presently being advertised in MacMall *and* Wired *magazine, indicates:*

"Acquisition time for high-definition images on a Power Mac is two seconds, on a Pentium machine it is five seconds."

From Michael E. Gaines
Pipeline Associates, Inc.
starman@powerpage.com

Why do I develop for the Mac platform? Experience. My career in computers started with PCs in 1985, but as the PC community grew, I didn't see the PCs themselves getting easier to work with. After a time, I got fed up with spending hours on incompatible hardware and moved to the Macintosh platform.

Working with the Macintosh computer has been a dream. Sure there are its ups, downs, and little tidbits that are hard to find documentation for, but as a whole, I'm happier on the Macintosh platform than on any other platform I've ever developed for.

The Macintosh Toolbox was the best idea Apple could have come up with. Not allowing any deviation from the rules is what sets development on the Macintosh computer apart from other platforms. Windows developers who I speak to have a tough time because Windows allows deviation from their core toolbox, which makes Windows development frustrating for them.

I'm comfortable here with the Macintosh family and I don't plan on giving it up for any other platform in the world.

• • •

Tell us why you prefer developing on/for the Macintosh computer instead of a Windows-based PC; your answers will be considered for use as part of this column. Send your contributions of approximately 250 words or less via e-mail to dreyfusp@apple.com.

less access to the Internet and/or an intranet, regardless of location."

More than 50 developers, including QUALCOMM, AllPen Software, and Lunatech Research, have announced plans to integrate the NIE into current and future releases of their products. "For the past two years, Internet connectivity efforts have been focused on the desktop environment," said Andrew Seybold, editor-in-chief of *Andrew Seybold's Outlook on Communications and Computing.* "The growth of Internet-based enterprise and commerce applications is driving an increased need for mobile Internet connectivity. High-functionality PDA platforms like Newton will play a significant role in providing that connectivity."

NIE is available immediately at no cost through the Newton developer home page at **http://dev.info.apple.com/newton**.

Apple Licenses Roaster for Java Support

Apple Computer recently announced that it has licensed the Mac OS–specific Roaster Virtual Machine (VM) and the associated "just-in-time" (JIT) compilation technology from Natural Intelligence. Apple will use these technologies to add support for Sun Microsystems' Java programming language to various Apple product lines. This announcement follows the April 30 announcement by Apple and Sun Microsystems that Apple has licensed Java.

At May's Worldwide Developers Conference (WWDC) in San Jose, California, Apple provided details on its plans to leverage the platform-neutral Java applet environment. Apple said it plans to embed Java across a range of Apple products and technologies, including the Mac OS, OpenDoc, Cyberdog, HyperCard, Newton, Pippin, and Apple Web servers. Additionally, Apple and third parties showed Java applets running inside OpenDoc parts.

Also at the WWDC, Natural Intelligence released an improved Developer Release 2 (DR2) version of its Roaster product. Roaster DR2 includes such improvements as support for 680x0 development, AppleScript and HTML support, technology for fetching and running Java applets off the Internet, and comprehensive, searchable electronic documentation. Purchasers of Roaster DR1 will receive the DR2 release automatically, as well as the next two releases, called Roaster Releases 1 and 2. Roaster DR2 purchasers will also receive Roaster Releases 1 and 2.

You can order Roaster from Natural Intelligence's Web site at **http://www.natural.com**.

First Pippin Appliance to Debut in U.S. This September

Bandai Co. Ltd., the first licensee of Apple's Pippin technology, has announced its intent to bring a Pippin product to the U.S. market well in time for the Christmas season.

Bandai Digital Entertainment, the U.S. subsidiary of Bandai Co. Ltd., recently announced that it would sell its @World Pippin-based product in the United States. The @World product is priced at $599 and is scheduled to be available in the U.S. in September 1996. The company is positioning it as a "multimedia and Internet information appliance."

Pippin-based products are powered by a PowerPC 603 processor and run an operating system based on the Mac OS. Thousands of titles already running on Mac OS–compatible computers can easily be modified to run on Pippin-based products, and any Pippin title can be played on any suitably configured Mac OS–compatible computer.

In his keynote speech at the Apple Worldwide Developers Conference, Apple CEO Dr. Gilbert Amelio said that Apple would consolidate each of its "appliance initiatives" (Pippin and Newton) to focus on providing Internet connectivity in the under-$1000 price range. The availability of Bandai Digital Entertainment's @World product in the U.S. market will be a visible sign of Apple's progress toward that goal.

SmartShelf: A New Way to Distribute Your Software

BITSource, a software distributor in Mountain View, California, recently announced a new electronic software delivery system that you can use to expand visibility—and sales—for your products. BITSource's new system is called *SmartShelf,* and your product can be included in it at no charge between now and August 31, 1996.

SmartShelf is an electronic software delivery system through which customers shop for software using an Internet-based catalog accessed from computers on a reseller's retail floor. Users browse through the online catalog and decide what products they'd like to order. Once users have submitted an order, the software is transferred to the reseller electronically, where it is written either to a recordable CD or directly to a hard disk. Corporate customers can obtain their own in-house version of SmartShelf through their associated reseller.

To help guide consumers through the often daunting task of selecting software, SmartShelf features a friendly, state-of-the-art electronic sales "wizard" called *The Software Guy.* When called upon, The Software Guy asks users a series of questions and then intelligently directs them toward the products that are most likely to be of interest, pointing out related special buys and "virtual bundles" along the way.

"At present, small and medium-sized resellers can't afford to carry a significant number of software titles," says BITSource President David Rubin. "Our goal with SmartShelf is to make it possible for these resellers to carry an unlimited number of titles. This will provide larger ISVs with incremental distribution into a market that until now has gone largely untapped, and it will provide smaller ISVs the shelf space they need to succeed."

The beta version of SmartShelf will be placed in all 19 stores of the Houston-based reseller Computize later this summer. Computize is one of the largest independent Apple dealers in the country. If you'd like your software to be included in the SmartShelf system, contact Rich Kline, director of publisher relations for BITSource (phone: 415-919-0652; fax: 415-919-0655; e-mail: rich@bitsource .com).

■ ■ ■

Apple Delivers Multiplatform LaserWriter

The LaserWriter 12/640, the newest member of Apple's LaserWriter printers, provides laser printing for users of Mac OS, Windows-based, and UNIX systems, and offers users a duplex printing option. The new printer is aimed at cross-platform small workgroups in areas such as business, government and education, where compatibility, duplexity, flexibility, and strong price/performance are required.

"The LaserWriter 12/640 PS is Apple's laser-printing solution for demanding small work-group environments," said Tom Mager, senior director of product engineering for Apple's Imaging Peripherals Systems. "Its flexible paper-handling options, extensive networking support, and Adobe PostScript and PCL printer language capability make it a natural choice for Mac OS, Windows, and UNIX users."

Featuring 600 x 600 dots-per-inch resolution enhanced with Apple FinePrint and PhotoGrade technologies, the LaserWriter 12/640 PS provides high-quality text and image reproduction at speeds up to 12 pages per minute. Media-handling capabilities include a 250-sheet universal cassette and an 80-sheet multipurpose tray, which supports media sizes from A4 to postcard, paper from 16 to 50 pounds, transparencies, and labels.

Autosensing, autoswitching Ethernet, LocalTalk, and bidirectional parallel ports are built in. Industry standard network protocols EtherTalk, TCP/IP, and Novell Netware IPX are supported through its Ethernet connection. Adobe PostScript Level 2 and PCL provide compatibility with cross-platform applications. The Apple LaserWriter 12/640 PS also ships with Adobe Type Manager for Mac OS and Windows, allowing users true "WYSIWYG" from screen to output.

The LaserWriter 12/640 PS supports a variety of paper-handling options. A duplex printing unit option allows for double-sided printing without a copy machine. Other options include the 500-sheet cassette and feeder for large jobs and an envelope cassette that handles a variety of envelope sizes and postcards. The face-up output tray option takes advantage of the unique straight-through paper path capability of the LaserWriter 12/640 PS; this is the ideal output option for printing on transparencies, labels, or heavier stock because it minimizes paper curl. Installation of the printer's paper-handling options does not require any tools.

The LaserWriter 12/640 is available immediately in the United States, Europe, Canada, Central and Latin America, Australia, and portions of the Far East. ♣

AppleDirections

Technology

CD HIGHLIGHTS

Tool Chest Edition, August 1996

This month's disc includes a small reorganization of the Sample Code folder: sample code is now organized by subject, similar to the Tool Chest folder. Also, you may have noticed that we've been able to squeeze a few more bytes onto recent CDs than we have in the past; this is thanks to our switch to Toast CD-ROM Pro 3.0, which does a much better job of optimization and System 6 desktop-file generation that our old methods. I'm not allowed to recommend a particular software package, but for the Developer CDs (wink, nudge) *we* use Toast Pro.

So, in addition to updates to the Arabic and Hebrew Language Kits, a Kanji version of System 7.5.3 (PCI only), and several localizations of Worldwide QuickDraw GX 1.1.3, here's this month's new and revised stuff.

Apple Game Sprockets 1.0GM

Apple Game Sprockets is a set of libraries that enhance your ability to create games for the Macintosh or any computer running the Mac OS. The four libraries in Apple Game Sprockets are SoundSprocket, DrawSprocket, InputSprocket, and NetSprocket. These libraries offer game developers several advantages over existing approaches to game development.

- SoundSprocket offers new technologies for presenting sounds localized in 3D space.
- DrawSprocket provides safe ways to manage the display and manipulate graphics, replacing the undocumented methods currently in use. It provides a consistent user experience for configuring game play. It also makes page flipping—long a capability of Apple hardware—accessible to games.
- InputSprocket provides a consistent user experience for configuring how input devices work with games.
- NetSprocket offers a common user interface for hosting and joining network games. Messages are sent and received across a variety of protocols with a simple programming interface.

Apple Game Sprockets alleviates the burden of tedious common details and allows you to concentrate on developing the game play. Developers of input devices also benefit from Apple Game Sprockets: It lets you concentrate solely on providing access to the device while leaving plenty of room for differentiation and adding value.

You can use the four libraries independently or together, as your game requires. These libraries are delivered to end-users as system extensions, similar to QuickTime or OpenDoc. For example, if your game requires the SoundSprocket library, you can include it as a system extension with your game. Apple Game Sprockets may be used without a license fee, and the SDK and run-time libraries are available on Apple's Web site at **http://dev.info.apple.com/games**.

C.S.M.P. Digests

This digest is a collection of article threads from the comp.sys.mac.programmer Internet newsgroups. It is designed for people who read the newsgroups sporadically and want an archive of the discussions. The C.S.M.P. Digest is moderated by Francois Pottier (pottier@clipper.ens.fr).

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

Macintosh PAP Client Interface

AppleTalk's Printer Access Protocol (PAP) is a session-level protocol that enables communication between workstations and printers or print servers. The Macintosh PAP Client Interface implements the workstation side of PAP for the Macintosh.

NetworkStream

NetworkStream is a C++ Stream IO class object for Open Transport Network I/O. Written in Metrowerks C++, it implements a TNetworkBuf, TNetworkOStream, and TNetworkIOStream object. For details, see the document ReadMe - NetworkStream.

Pippin Technical Notes

Pippin is a new multimedia and Internet playback architecture based on Power Macintosh and licensed by Apple to Bandai Digital Entertainment. Because it is based on the Macintosh architecture, most Pippin development is the same as Macintosh development. The technical notes in this folder cover all the Pippin-specific aspects of developing on Pippin.

QuickDraw GX Sample Code Update

This folder includes replacements and updates for the sample code folder of the QuickDraw GX SDK.

Snippets Update

This folder includes the following snippets:

- *PDlog Expand (Printing).* This example shows how to add items to print dialog boxes (with Balloon Help and "subdialogs") as described in Macintosh Tech Note PR09.
- *PutAwayVolumes 1.1 (Files).* This snippet demonstrates several techniques for unmounting a volume.

Word Services Suite 1.0.7

The Word Services Apple event suite allows any application to link to a spelling checker, grammar checker, or other text service as if it were a built-in menu item. This is probably the simplest useful thing you can do with Apple events.

The Word Services SDK contains the suite specification, tutorial documents, the

please turn to page 24

Apple’s New Image-Capture Platform

Opportunities to Create Computers That See and Cameras That Think

By Kris Newby, Apple Directions *staff*

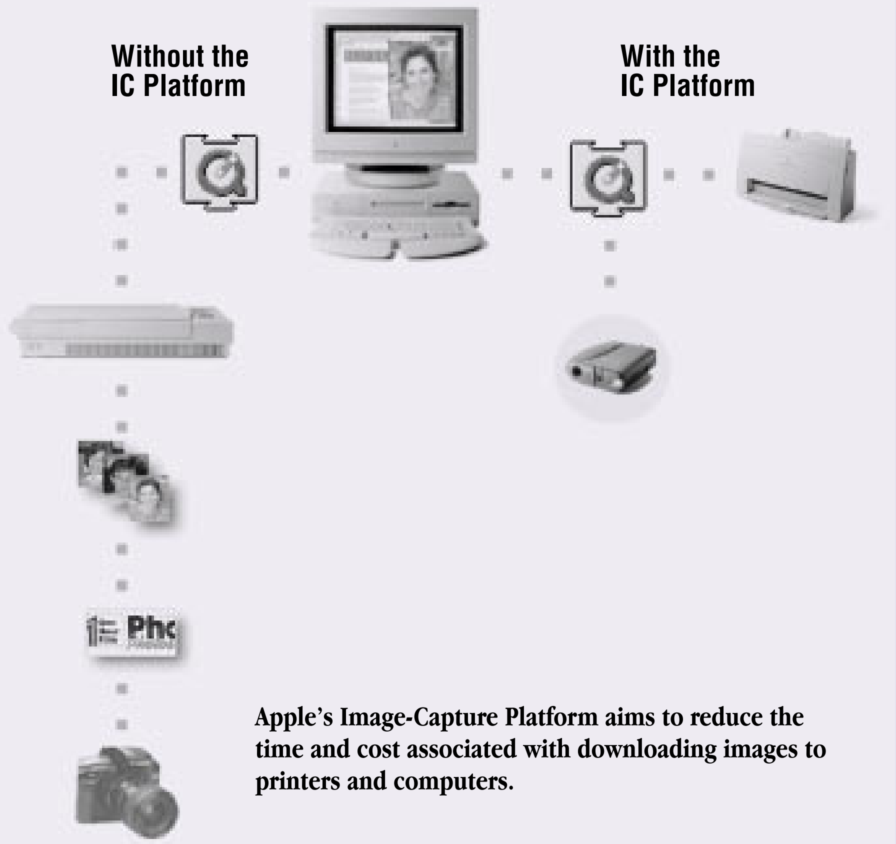
There’s a funny thing about photos—almost no one ever throws them out. We all have boxes of them, tucked away in dark places. And though most of these photos are blurry and uninteresting, it’s hard to part with them. They seem to represent something more than little pieces of paper coated with light-sensitive ink—their value lies in the fact that they provide us with one of the best ways to communicate concepts, memories, and emotions.

Because of the compelling nature of visual images, Apple’s new Image Capture (IC) Platform offers you a number of business opportunities for enhancing your current products and creating new ones. Looking at this platform from a developer’s perspective, Apple’s IC Platform consists of two main components: A comprehensive set of QuickTime IC application programming interfaces (APIs) and a multitasking, real-time embedded operating system, the QuickTime IC operating system, which runs inside digital cameras and devices equipped with the Motorola MPC823 or MPC813 microprocessor.

The QuickTime IC APIs provide a bridge between digital imaging hardware and software applications, making it easier for software developers to build imaging-related features into new and existing products. (See “Apple Image-Capture Platform Benefits” on page 18). The QuickTime IC operating system will help imaging device manufactures create “smart imaging appliances” with LCD (liquid crystal display)interfaces and on-board image-manipulation capabilities.

Looking at this technology from a market perspective, Apple’s IC Platform will catalyze the creation of easy-to-use “computers that see and cameras that think,” says Dan Torres, the Apple image capture product marketing manager. This article provides you with an overview of the technology and business opportunities associated with the Apple IC Platform.

Apple’s Image-Capture Platform



Apple’s Image-Capture Platform aims to reduce the time and cost associated with downloading images to printers and computers.

A Birds-Eye View of Imaging

The need for faster, better, and less expensive ways to integrate images into multimedia products, World Wide Web sites, and television has become painfully obvious. The traditional process of capturing and processing images—exposing film, getting it developed, scanning in selected images, downloading them onto a computer, then using specialized software to edit the images—is cumbersome at best. Digital technology offers some marked advantages over paper solutions: Digital images are more easily resized, edited, stored, and transported electronically. But up until now, the adoption of digital image processing has been impeded by the cost of the imaging devices and the lack of an industry-standard software bridge between these devices and computers.

“Traditional photography has advanced to the point where it is inexpensive, high quality, and virtually foolproof,” says Don Strickland, president and CEO of PictureWorks (and also a former VP of Apple’s Imaging Group.) “This is the result of cameras, film, and processing all working together as a system. In the case of digital cameras, a systems approach has not yet been realized. Today’s digital cameras operate more as stand-alone peripheral devices designed to capture images and download them to a host computer. Apple’s QuickTime IC APIs make it possible, for the first time, for digital cameras to become part of an integrated digital image-capture system that spans the camera, host computer, output destination, and application software. The impact on the digital camera users will be phenomenal.”

The IC Platform is a natural extension of Apple’s leading multimedia and Internet authoring platform. Apple already provides developers with the best platform for integrating images, video, animations, and audio media on computers, and now the IC Platform will make it easier to get images into

please turn to page 17

HUMAN INTERFACE

Designing for a Complex World Intelligence

By Peter Bickford

[Note: Pete Bickford is on vacation, so we're reprinting one of the very first articles he wrote for this column, way back in July 1992. We like to think of this as one of Pete's greatest hits. We hope you agree.—the Editors]

In our everyday life, we carry on countless conversations with people who mispronounce words, use incorrect grammar, or have strange accents. Most of the time, we have no trouble understanding what the other person is saying, even though their speech doesn't exactly follow "the rules." These minor "errors" of speech are so common that we don't usually notice them—indeed it's considered rude to point them out to the speaker. Only in the rarest of instances will we ask the speaker to restate something that we don't understand.

Unfortunately, no similar attempt at understanding seems to take place in the world of computers. Instead, we have written a generation of computer programs so that they behave like officious bureaucrats: "comma required between items," "semicolon required after previous line," "first letter of name must be capitalized." These are the messages of a computer that wants to be shot. Users who read messages like those will conclude that the computer is either rude or stupid—and they'd be right. It's up to us to be sure the Macintosh is always thought of as a kinder, gentler, and more intelligent computer: it's one of the things that keeps all of us from going the way of Brand X.

Our applications need to start behaving with more intelligence. *Intelligent systems* are ones that work hard to understand our input, assist us in arranging and formatting our work, configure themselves to match the computing environment, and (ideally) take care of repetitive or housekeeping chores for us, leaving us free to do our real work.

Basic Intelligence

For starters, don't let your application be pedantic about trivial input formatting rules. If the computer knows that the first letter needs to be capitalized, it should simply adjust for it, then move on. People who point out others' trivial grammatical errors can have trouble making friends. Applications that do the same can expect similar difficulties.

For simple cases when the user's input doesn't conform to a formatting rule, you can usually solve the problem by converting code that flags such errors to correct for them instead. When the correct interpretation of a user's input is not quite so obvious, design your application to take its best guess, then ask the user if that was what he or she meant. For instance, if the user enters a customer name that is unknown to your system, have the program offer a list of names similar to the one typed. Ideally, this list would include names that both vary from the one typed by a letter or two (*Anderson* vs. *Andersen*), as well as ones that vary greatly, but that are phonetically similar (*Anderssonne*). Of course, the user should also be able to pick from a complete list of all names in the system (the "see-and-point" instead of "remember-and-type" principle).

Having the computer make formatting corrections needn't just be a way to prevent error; it also lets the computer help the user put things in the form that they really wanted in the first place. Draw programs can provide the option of automatically aligning objects, scanning programs can straighten images, and personal information managers can format text. In all of these cases, you can design applications that can formulate a good idea of what the user wanted in the first place, and apply a few simple rules to transform their input into that form. Of course, the user should also have the option to turn this special formatting off.

Delight the User

Beyond these simple formatting changes comes the idea of anticipating the user's needs and having your application adjust its defaults accordingly. A nice example of this is the way a popular drawing application for Macintosh lets you quickly create a series of objects. As with most applications, if you duplicate an object in this application, the copy appears just below and to the right of the original. The really neat part, though, is that if you move the copy, say three inches above and to the right of the original, then press Duplicate again, the next copy will appear three inches above and to the right of the last one. Since the program took note of the user's actions and adjusted itself accordingly, it now becomes incredibly simple to create a line of evenly spaced objects.

Another application I admire is the address book application I use. Like most such applications, this one contains a phone dialer, which keeps a record of my local area code so that it knows when to dial long distance. However, the application also realizes that if I enter someone's phone number and I omit the area code, it's more than likely a local call. As such, it formats such numbers with my local area code, which I'd entered in the phone dialer. Sure it's a simple thing, but just like the object alignment in the drawing program, it's one of those features that delights users with its thoughtfulness the first time it's discovered.

Help the Computer Know Itself

The intelligent application never asks the user for information it can find out for itself. Macintosh applications have a real advantage in this regard, as they can use the Toolbox to detect a great deal about the operating environment, including the type of Macintosh being used, the operating system features that are available, whether the monitor is color or black and white, and much, much more. By using this information, you can have your application behave appropriately for its environment without quizzing the user on configuration.

Although we've made a good start in having our applications be self-configuring, we can still do more. For instance, the first time a terminal program is used, it can conduct a series of tests to determine whether a modem is connected, what port it's connected to, and the speed at which it's capable of operating. The results of this testing can be used as defaults when the setup is first presented. If the tests are inconclusive, nothing is lost by having tried—but with any luck, such testing can provide much of the information needed for the user to make a connection on the first try.

Intelligent Agents: Computerized Assistants

Prior to System 7, the Macintosh System Folder was a mess—home to hundreds of control panels, extensions, preference files, and other miscellaneous items. System 7 cleaned up the mess by making dedicated subfolders for these different types of files, but this introduced a new problem: now the user had to make sure that items wound up in the right subfolder. This problem was solved by adding intelligence to the Finder, which allowed the user to merely drop the different items on top of the System Folder; they would be automatically placed in the proper subfolder. Because the system knows where the different types of files belong, it doesn't make the user go to the trouble of placing them manually.

As our systems become more complex, there's a greater need for these types of facilities, which free the user from mundane housekeeping tasks. Eventually, intelligent, automated processes called *agents* may become computerized assistants, backing up a hard disk, scanning new services for articles of interest, and sorting mail while users concentrate on their real work. In the meanwhile, we need to look for any opportunities to use our knowledge of the user's actions, and the particulars of the task at hand to automate trivial or repetitive processes.

Be Specific and Don't Go Too Far

When creating intelligent interfaces, you should bear in mind a couple of issues. First, your interface can only behave intelligently if it has specific information about the data being worked with—and the more knowledge it has about it, the more intelligently it can behave. For instance, it has to know what a phone number is, as opposed to just a collection of integers. Similarly, you're limited in the amount of intelligence you can bring to bear on a "customer name" field if all your program knows about names is that they fit in a 30-character alphabetic field. You'll need to build the specific knowledge of how to deal with various types of data in your application—while at the same time taking steps not to hard-code yourself into a corner.

Secondly, you need to know when to stop. No matter how valiantly your telephone number parsing algorithm tries, someone out there is going to enter a number that doesn't seem to fit, for instance an international or alternate long-distance number. The same holds for names and addresses. And yes, sometimes you really didn't want to have your objects aligned in your drawing program. In all these cases, allow the user to override, or even turn off, the intelligence features. If computers are ever to become our assistants, we'll definitely need to raise the level of human–computer communication—but even in human conversation, you sometimes do need to (politely) ask other people to rephrase what they said.

Peter Bickford is a member of Apple Computer's Human Interface Design Center. You can reach him on the Internet at bickford@apple.com.

TECHNOLOGY FEATURE

IC Platform

continued from page 15

computers, by enabling direct image downloads from IC-savvy devices and easier manipulation of those images from within applications. Leveraging its experience with the QuickTime cross-platform standard, Apple is working with image-capture device vendors, developers, and other computer manufacturers to help define and establish this new set of protocols. This in turn will help Apple achieve one of its goals—to bring easy-to-use, high-quality images to the widest possible range of users, from graphic designers, to webmasters, to amateur and professional photographers. (See the figure on page 15 to see how the IC Platform will change image processing.)

The IC Platform Operating System

At the lowest level of the IC Platform is the multitasking, real-time QuickTime IC operating system that runs on top of a Motorola MPC823 or MPC813 chip. (See the figure on page 19 for an overview of the IC Platform architecture.) This operating system will reside inside future image-capture devices, such as Apple's next-generation QuickTake camera, third-party digital cameras, video cameras, and scanners. Though it's too soon to say who will build this operating system into their products, most major camera companies—including Sony, Minolta, Olympus, Panasonic, Nikon, Casio, Fuji, Kodak, and Agfa—have met with Apple to learn more about it. (*Apple Directions* will keep you informed of the latest IC Platform hardware adopters as they publicly announce their support.)

Today most high-end cameras are embedded with an integrated circuit that provides users with five or six control commands (flash, f-stop, and so on). Apple's new QuickTime IC operating system will provide these manufacturers with a software toolbox of more than 200 commands, as well as a more scalable, extensible architecture. This in-camera software enables features such as

- standardized access to traditional camera settings
- cross-platform DOS and QuickTime file support
- a Mac OS–like graphical user interface
- support for JPEG image compression, and probably MPEG compression, for use in IC-savvy video cameras (see the box on page 20 for definitions of the imaging terms used in this article)
- file handling for downloading images to a computer's RAM, flash memory, or hard drive
- multiple communication protocol support, enabling capabilities such as direct image downloads to an output device or the Internet
- audio, text, and graphics support
- commands for automating time-lapse and VR image captures
- scripting support
- a localization manager
- customizable menus for access to unique hardware functions

Bottom line, use of this architecture will result in smarter cameras, and soon you'll see cameras such as these:

- *Cameras that record sound.* QuickTime IC includes support for audio (11.025-kHz mono to 44.1-kHz stereo), enabling users to record "sound bytes" along with images.
- *Cameras that annotate images with text and graphics.* IC-savvy cameras will be able to

output images annotated with graphics, complex fonts, and multiple languages. With these types of features, you'll see digital cameras used to do things such as instantly printing employee badges with an employee's photo, name, and company logo.

- *Cameras with LCD graphical interfaces.* Some IC-savvy cameras will include displays that enable users to instantly preview, edit, print, and share images. To present these advanced features, QuickTime IC cameras support a Macintosh-caliber graphical user interface, designed to intuitively guide users through a camera's features.
- *Cameras that perform scriptable functions.* Because all QuickTime IC cameras include a scripting function, you'll be able to write scripts that automate tasks such as "stitching" together still shots into QuickTime VR panoramas or linking time-lapse photos into QuickTime movies.

On the camera side of the imaging business, there are numerous benefits to this open standard. First, by adopting the IC Platform, these camera manufacturers are going to sell a lot of digital cameras. Although many vendors will have to reengineer an existing line of cameras, this new architecture provides them with hundreds of new ways to differentiate their products. By moving from hardware-based features to more software-based features, they'll reduce the time-to-market of next-generation devices. And by interfacing to a common set of driver APIs, they'll significantly reduce the effort required to write device drivers for new image-related applications.

The Host Software

For users to access IC Platform features, their computers will have to run QuickTime IC host software. In the near term, this software will be delivered to users through software extensions; eventually the software will ship as part of the standard QuickTime cross-platform solution set.

Within the digital scanner industry, TWAIN is the dominant scanner-to-computer set of data-transfer APIs, though many developers complain that this standard is too difficult to implement and validate. (This cross-platform specification is maintained by the TWAIN Working Group, which includes Hewlett-Packard, Logitech, Documagix, and Canon.) The QuickTime IC Platform goes beyond this standard, addressing the needs of multimedia developers and camera vendors. But because QuickTime IC includes TWAIN support, images acquired through TWAIN-compliant devices will be able to be read by QuickTime IC–savvy applications.

QuickTime IC APIs

The QuickTime IC API set, in essence, is an extension of QuickTime, Apple's multimedia architecture for displaying and storing time-based media on Mac OS and Windows-based computers (and eventually other platforms). Any application that supports the top-level QuickTime IC APIs will allow users to view and perform simple edits on any digital image captured by an IC-savvy device. And since it's part of QuickTime, QuickTime IC host software also provides users with the ability to access

Apple Image-Capture Platform Benefits

	Features	Benefits
Imaging hardware developer benefits	Built-in MPC8xx chip and QuickTime IC operating system, offering more than 200 commands.	Facilitates the development of smarter cameras and imaging devices. Decoupling new features from hardware enables faster innovation and a core-competency focus.
	Open, extensible, cross-platform Camera Interface APIs.	Saves the time and money required to write hardware-specific drivers for each new imaging-related software application.
	Easy downloading of digital images to computers or output devices.	Motivates mainstream consumers to purchase digital cameras and color printers.
	Extensible, customizable Panel APIs.	Enables specialized digital camera uses, which in turn will sell more imaging devices.
Software developer benefits	Panel APIs with a consistent interface for capturing, previewing, and editing images.	Provides an easy way to differentiate software products with image-manipulation capabilities.
	Open, extensible, cross-platform Camera Interface APIs.	Reduces the effort required to support cross-platform driver software for popular digital imaging devices.
	Lower cost and complexity to work with digital images.	Lowers development costs for multimedia developers. Opens a new category of mass-market software applications, from electronic photo greeting cards to low-end photo-editing applications.
	Extensible, customizable Panel APIs.	Generates business opportunities for creating custom or vertical market plug-in panels—for VR scene stitching, color matching, photo IDs, simple video conferencing, and so on.

PICT, JPEG, QuickTime movie, and BMP media.

QuickTime IC APIs will save time and money for software and hardware developers who work with image-related products. Hardware developers won't have to write (or pay an outside group to write) device-specific drivers for each new product they release; instead, they'll be able to write to the standard Camera Interface APIs, and this software layer will take care of the device-to-application data transfer.

QuickTime IC's top API layer, called the *Panel APIs,* will consist of six simple commands. With QuickTime IC host software running, users can access interface panels that enable them to control a camera, preview images, and perform simple image edits. (These interface panels are shown on page 20.) Because of QuickTime IC's modular architecture, camera vendors and developers can also add custom panels. The three main panels that will ship with QuickTime IC include the following:

- *The camera panel.* This provides a graphical user interface that enables a user to control a camera's shutter speed, aperture size, flash, and focus.
- *The viewer panel.* Users are able to use this interface to preview, arrange, and discard images.
- *The PICT panel.* These editing tools enable users to crop and adjust the lightness and darkness of digital images.

The Camera Interface APIs also enable direct developer control of this rich set of camera commands, enabling them to create plug-in panels that extend device functionality with photo-optical mainpulation (similar to Adobe™ Photoshop filters), scripting, grouping, tagging, custom image processing, and more.

The consistent user interface that this API set provides will be a boon to users. For instance, a photographer with a case full of cameras will be able to quickly switch between different camera models, just as Mac OS users are able to move between software applications with ease. Developers will also be able to market "plug-in" panels to digital camera users, offering, for example, a panel that automatically adjusts image color for fluorescent lighting.

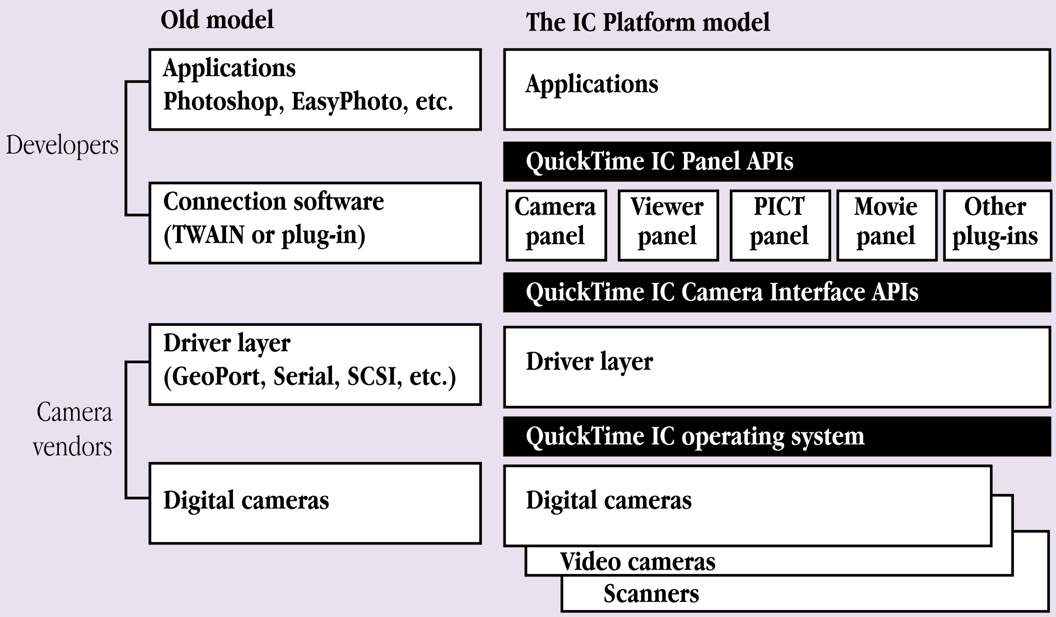
Developer Opportunities

At a minimum, QuickTime IC presents an opportunity for adding image-manipulation features to software applications. By supporting the Panel APIs, your users will be able to work with digital QuickTime IC images within your application. And as additional APIs are released in the coming year, you'll be able to add more advanced, value-added image-manipulation capabilities.

Andy Parng, president of QuickMedia, recently discussed one of the QuickTime IC features that he'll be adding to one of his products: "QuickTime IC will enable our users to download images into our Internet-based photo album, Living Album, directly from any compliant digital camera or scanner."

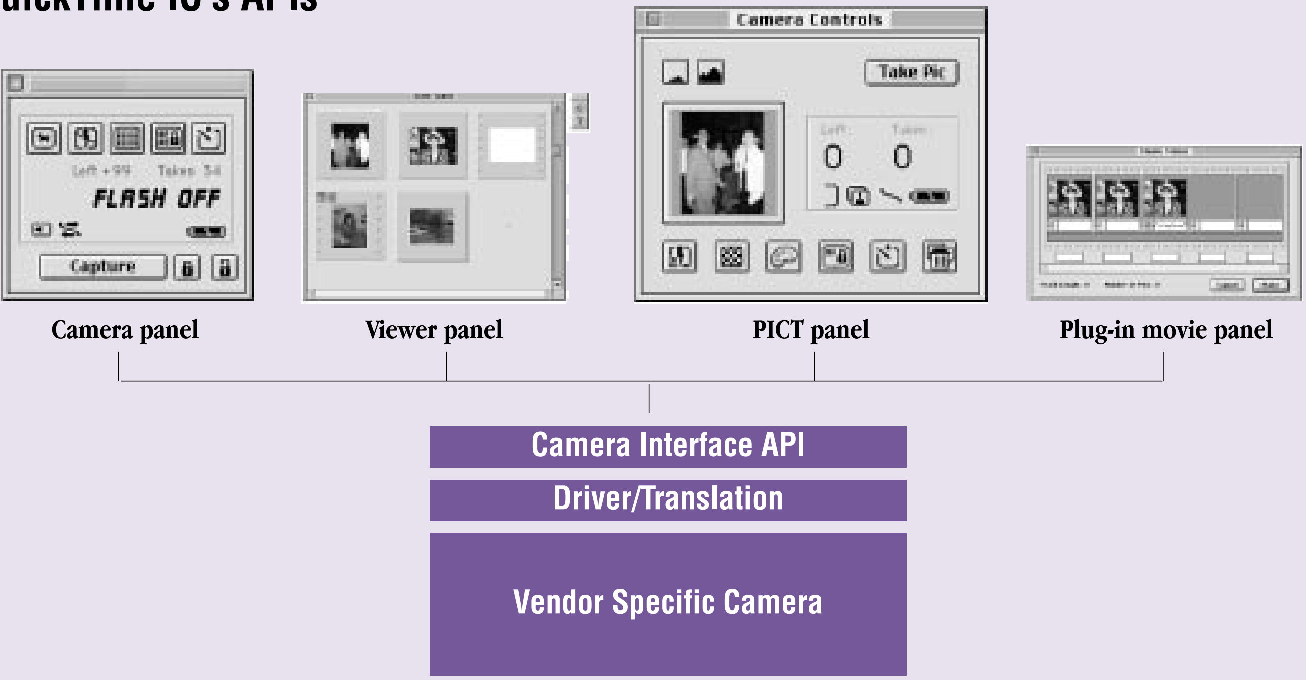
Even if your applications are outside of the

The Apple Image-Capture Platform Architecture



The Apple IC Platform consists of the *QuickTime IC operating system, QuickTime IC Camera Interface APIs,* and *QuickTime IC Panel APIs.* The QuickTime IC operating system runs inside digital cameras and devices embedded with a Motorola MPC823 or MPC813 microprocessor. It provides hardware developers with more than 200 commands to help them create "smart imaging appliances" with advanced functions. The Camera Interface APIs provide users with a standard interface for controlling digital cameras from on-board displays or computers. These APIs also save hardware developers the cost of driver development for their imaging devices. The Panel APIs provide software developers with some simple commands that make it easy to add standard and custom image-manipulation capabilities to their software applications.

QuickTime IC's APIs



QuickTime IC's APIs provide digital camera users with a consistent user interface for controlling, previewing, and editing images on an LCD camera display or a computer.

multimedia category, you may find that QuickTime image support offers a good way to differentiate and add value. For example, a developer of a personal information manager could add a feature that lets users drop in contact photos. A developer of a media database could use the QuickTime IC and ColorSync APIs to let users find and sort images by dominant color.

The advent of "smart" cameras will create opportunities for you to create specialized solutions that help photographers and graphics professionals save time and money. For example, you might develop a panel that helps digital camera users with exposure bracketing—taking photos over a range of exposure settings to ensure that the best image is captured. Another panel could help photographers automate QuickTime VR lighting adjustments and scene "stitching" within a digital camera.

This technology will also catalyze the creation of innovative new solutions such as these:

- The in-camera time-lapse API could be combined with a simple QuickTime movie creation tool for multimedia presentations and stop animation sequences.
- A systems integrator could combine an IC-savvy camera and a microscope system to create an automated pathology lab diagnostic tool.
- QuickTime IC's built-in communications could be used to create a live image feed from a digital camera to a Web page. For example, you could create an inexpensive Web-based monitoring system for traffic conditions.
- You could create a script that uses QuickTime IC calls to help real estate companies quickly post VR home walk-throughs to their Web pages.
- With the ability to automatically place text and graphics directly on an image, you could develop a solution that instantly creates photo IDs, conference badges, or business cards. (IC-savvy cameras support direct print capability.)
- By combining a QuickTime IC camera with a customer panel interface, developers could create an interactive and intuitive onboard camera user guide.

A Focus on Consumers

In a recent focus-group study, Apple took pictures of test subjects using a QuickTime IC–based camera, then asked subjects their opinions of the medium-resolution color ink

IC Platform Resources

Here are some Internet resources to explore if you want to learn more about Apple's new Image Capture Platform and related imaging standards.

- Apple IC Platform press release (May 13, 1996)—**http://product.info.apple.com/pr/may.html**
- Motorola MPC823 and MPC813 microprocessor for imaging peripherals—**http://www.mot.com/SPS/PowerPC/library/press_release/823.html**
- A list of digital camera vendors—**http://www.cgw.com/cgw/vendors/hardware10.html**
- TWAIN industry standard protocols for scanning devices—**http://www.twain.org/www/whtpaper.html**
- To check out some cool imaging solutions, see the Web sites of QuickMedia Labs (Web-based photo album)—**http://www.quickmedia.com**; PictureWorks (image editing software)—**http://www.pixworks.com**; or Storm Software (mass market snapshot scanner)—**http://www.easyphoto.com/storm**.

jet print output. The response was lukewarm at best. *But,* when shown the same image cropped and printed in a decorative card format (suitable for a letter to grandma or refrigerator display), the subjects were thrilled. This study drove home the importance of creating tools to help consumers easily edit and share images.

"QuickTime IC could signal a tremendous breakthrough for customers," says Kristy Holch, president of MegaTrends Research. "It's akin to the 1985 introduction of the laser printer equipped with PostScript™. The printer provided the environment for desktop publishing when combined with other items such as fonts and application software."

Creating tools that help mainstream computers create and share images probably represents the most lucrative long-term business proposition to developers. Today, just about everyone has at least one camera and takes an incredible number of photos annually. To get a feeling for the potential, consider this: In the book *CD-ROM: Publishing Medium—Publishing Tool* (ISBN: 0941845125), author Les Cowan states that if only 1 percent of all the photographs developed each year were transferred to Kodak Photo CDs, they would fill 5 million discs. And by the year 2000, International Data Corporation estimates that there will be 2 to 5 million digital cameras in the hands of consumers, photographers, and service bureaus. That represents a respectable market opportunity if you can grab a piece of it.

QuickTime IC opens up application possibilities that today are only accessible to multimedia developers and graphic artists. Screen savers that display your favorite photos or photo-enhanced Internet greeting cards will be as commonplace as clip art packages. The first developers to get those amateur photo images out of shoe boxes and into the hands of friends and relatives will be the market category leaders of tomorrow.

Availability and Timing

Today all the QuickTime IC APIs are defined and functional, and first-generation imaging hardware is operational. The DR1 release of the Panel APIs will be available in July or August of 1996, followed by the release of a final SDK later in the year. Apple is actively collaborating with digital camera vendors at this time, and based on their feedback, the Camera Interface and OS-level APIs will be released in stages during 1997.

QuickTime IC APIs can be used by any developer, though Apple will maintain ownership of the underlying code. To use this code, you'll have to fill out a licensing agreement, similar to the QuickTime agreement, but no licensing fees or royalties will be required. If you need more information about QuickTime IC or licensing, contact Dan Torres (t2@applelink.apple.com) or Steve Saylor (saylor1@applelink.apple.com) at Apple.

Though Ansel Adams, the famous American photographer, didn't own a digital camera or a Macintosh, he recognized the value of timing in the image-capture business when he said, "Sometimes I get to places just when God's ready to have somebody click the shutter."

Timing is important in an emerging technology market such as digital imaging. Enter the market too early, and you risk adopting the wrong technologies and standards. Enter too late, and capturing market share becomes exponentially more difficult. With Apple establishing the first camera-to-computer standard, and with recent price drops in digital cameras and color printers, the timing is finally right for enterprising developers who want to enter the image-capture market. ♣

Kris Newby (newby.k@applelink.apple.com) is the business editor of Apple Directions *and a freelance writer based in Palo Alto, California.*

Imaging Glossary

The image-capture industry uses an alphabet soup of acronyms to describe data formats and protocols. Here are definitions of some of the terms used in discussions about QuickTime IC.

• *BMP.* This is the common bitmap graphic file format used with Microsoft Windows and IBM OS/2 operating systems. QuickTime IC host software provides users with the ability to access BMP media.

• *FlashPix.* This new standard will help users with one of the challenges of digital imaging—working with large file sizes. This hierarchical file format enables users to quickly extract multiple-resolution images from a given file, so they can work with "memory-lite" placeholder images. This standard is being backed by Microsoft, Live Picture, Apple, and other companies.

• *Image Pac.* This is the commercial name for Kodak's open compression-decompression (codec) file format that's used for storing images on Photo CDs. (Photo CDs "hold" about 100 images.) Mac OS uses QuickTime APIs to read these files.

• *MPEG.* Right now it looks as though QuickTime IC will support MPEG files, a standard for digital video compression established by the International Standards Organization's Moving Picture Expert Group (MPEG). Because MPEG delivers an image that is far superior to competing methods, and because it's an open standard, MPEG-compressed data streams are likely to be supported by most emerging interactive video services.

• *JPEG.* Because of its small file sizes, many people use the Joint Photographic Experts Graphic (JPEG) compression format for transferring images by modem or posting on the Internet. This compression scheme actually throws away data to save space, so you'll notice image degradation if you opt for a high level of compression. QuickTime IC supports this format.

• *PICT.* This is the most commonly used Macintosh bitmap graphics format. Apple's version of PICT is a QuickTime-compatible format that includes image tags, thumbnail images, and JPEG compression, as well as QuickTime extension and audio support.

• *SanDisk's CompactFlash.* This is the 1.7 x 1.4 x .13 inch card standard for the storage and transfer of digital image files. It's being supported by Apple, Canon, Kodak, H-P, Polaroid, Seagate, and Seiko.

• *TIFF.EP.* QuickTime will support the new extended version of the Tagged Image File Format (TIFF.EP). This format works well for the printing and on-screen display of photographic images.

• *TWAIN.* This is the dominant scanner-to-computer set of data-transfer APIs that QuickTime IC will, in part, use to achieve Windows compatibility. (This acronym, by the way, is supposedly short for *Technology Without An Interesting Name.*)

OpenDoc Human Interface FAQs

By Kerry Ortega, Dave Curbow, Elizabeth Dykstra-Erickson, and Geoff Schuller, Apple OpenDoc Human Interface Team

Common Problems Observed in Real-World Parts

Recently, we've reviewed several part editors that went on the OpenDoc DR5 CD. This month, instead of presenting question-answer pairs, we've made some observations based on the problems we saw with various parts. To point you to more information on the topic, we've also included page references to the *OpenDoc Programmer's Guide.*

• *An embedded part should not change the Document menu.*

Items in the Document menu apply to the document as a whole. When your part editor is the root part, it may add commands to this menu. However, these commands *must* also apply to the whole document. (For example, a part might add a Fax command that allows users to fax the entire document.) Think carefully before adding commands to this menu. If you must, add them at the bottom, after the printing commands.

If your part editor is an embedded part, we highly recommend that it never add any commands to the Document menu. Since this menu's commands apply to the document, users will expect them to be available all the time. Because any commands that your embedded part adds will only appear when your part editor is active, users will have to remember which parts to activate to access particular Document menu commands. This could be very confusing for new users.

• *Your container part editor's Insert command should work when it is enabled.*

Users choose the Insert command to select a document through the standard file dialog box and embed or incorporate it into their documents. Your part editor should follow the same steps for embedding or incorporating intrinsic content. This may not be clear in the "Human Interface Guidelines" section of the *OpenDoc Programmer's Guide.*

When using the Insert command, you should add the content at the current insertion point in your part. If there is a selection instead of an insertion point, add the content *after* the selection. This behavior is different from that of a paste operation, but is consistent with what happens when users drag content into a document. In both cases, users shouldn't lose any previously selected content. At the completion of the insert or drag operation, the part editor should select the new content. As mentioned in the guidelines section of the *OpenDoc Programmer's Guide,* there is an exception: Text should not be selected. Instead, the part editor should place the insertion point after the new content.

If your part has neither an insertion point nor a selection, your editor should place the content in the middle of the active frame or at some other default position that makes sense for your part editor. If the document is read-only, the Insert command should be disabled.

For more information, see pages 249, 344, 347, 541, 551, 594, and 595 of the *OpenDoc Programmer's Guide.*

• *Support both the Print and Page Setup commands.*

The root part of the window handles both the Page Setup command and the Print command. Root parts are responsible for overall document features such as page size, as well as the printing behavior of their documents. In general, the printed version of your part should be a faithful representation of its screen appearance. If the part displays scroll bars, your editor typically prints them, unless your part is the root part. You also should only print the visible portion of the content in your frame.

The Command-P key combination is reserved for printing. Do not use it for anything else.

For more information, see pages 250 and 542 of the *OpenDoc Programmer's Guide.*

• *Support the Undo/Redo commands.*

One of OpenDoc's most useful features is support for multiple levels of undo capability, which your part editor should support. If you support only a single level of undo capability, other parts that support multilevel undo capability will lose their undo history.

Your part editor determines the actions to undo. Actions that take a long time to re-create are probably actions users would expect to undo. In general, support undo functionality for the operations that change content in your part. Operations that your editor doesn't have to undo include scrolling, selecting, opening, closing, or changing the position of a window or of icons within a window; none of these operations change content.

Reserve the Command-Z key combination for the Undo command and Command-R for the Redo command. You should not use these key combinations for any other purpose.

For more information, see pages 261, 544, and 590 of the *OpenDoc Programmer's Guide.*

• *Support the Paste As command.*

The Paste As command allows the user to control how the part editor pastes Clipboard content and whether to create a link. Even if your part editor does not support linking, users can still control the merge/embed decisions and the kind of content they want pasted. If your part supports neither embedding nor linking, your part editor should disable Paste As in the Edit menu. Your part editor should also disable Paste As if the document is read-only.

For more information, see pages 545, 551, and 597 of the *OpenDoc Programmer's Guide.*

• *View in Window should show all the part's content.*

The View in Window command opens the active part into a separate part window so that the user can edit the part's content as a whole. You should make the part window large enough to display all of your part's content, not just the content showing in the source frame.

The part editor should open the window at an offset of 20 x 20 pixels from the upper-left corner of the frame that is the source of the part window. If the current script is right-to-left rather than left-to-right, open the window from the upper-right corner instead.

For more information, see page 573 of the *OpenDoc Programmer's Guide.*

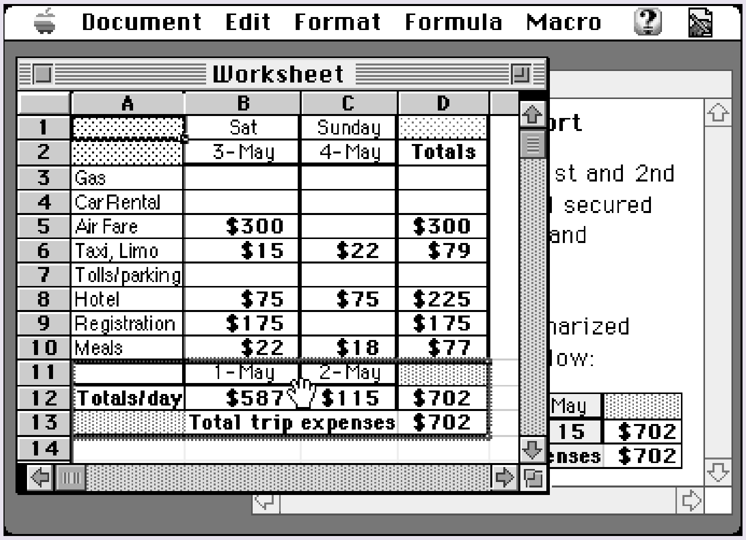
• *Support the Show/Hide Frame Outline commands.*

When the user opens a part into a window and the part's windows display more part content than is visible in the source frame, your part editor should add a Show Frame Outline command to its Edit menu. This command allows the user to reposition the content that is visible in the part's source frame. This command also creates a mode in which the user can only reposition the visible content or exit the mode. When the user chooses Show Frame Outline, your part editor displays a border around the content currently displayed in the source frame. This border represents the source frame.

When the user moves the mouse pointer within the border, the pointer changes into an open hand. (See the figure "Manipulating the Show Frame Border.") When the user presses the mouse button, change the open hand into a closed hand. When the user drags the pointer, move the border, tracking the pointer position until the user releases the mouse button.

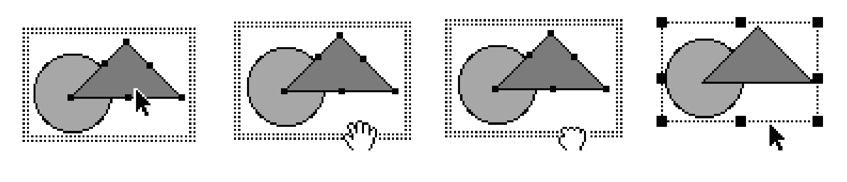
For more information, see pages 548 and 575 of the *OpenDoc Programmer's Guide.*

Manipulating the Show Frame Border



	A	B	C	D
1		Sat	Sunday	
2		3-May	4-May	Totals
3	Gas			
4	Car Rental			
5	Air Fare	$300		$300
6	Taxi, Limo	$15	$22	$79
7	Tolls/parking			
8	Hotel	$75	$75	$225
9	Registration	$175		$175
10	Meals	$22	$18	$77
11		1-May	2-May	
12	Totals/day	$587	$115	$702
13		Total trip expenses		$702
14				

Manipulating a Part



This figure shows four views of a part and the mouse pointer. In the first view, the mouse pointer is a normal pointer, and the user can change the part's content. In the second view, the pointer changes to an open hand when it moves within the part's border. In the third view, when the user presses the mouse button, the pointer changes to a closed hand. In the fourth view, when the user releases the mouse button, the pointer changes to the normal pointer and the part itself becomes selected.

• *Carefully select your part's category name.*
If your part fits into one of CI Labs' defined categories, use that for your category name. If not, check with CI Labs (cilabs@cilabs.org) about creating a new category if there isn't an appropriate one for your part.

For more information, see pages 71 and 476 of the *OpenDoc Programmer's Guide.*

• *Show the "open hand" pointer when the mouse is over an active border.*
To help users know that the mouse pointer is over a part's active border, the part editor should change the pointer to an open hand. If the user then presses the mouse button, the open hand changes to a closed hand. When the user releases the mouse button, the part is no longer active—it is selected, which means that the user can now manipulate the part as a whole. (See the two middle views of "Manipulating a Part.")

Using these pointers will make selecting parts much easier, since users won't have to guess when the "hot spot" of the pointer is over the active border.

For more information, see page 536 of the *OpenDoc Programmer's Guide.*

• *Support the selection modifiers (for example, Shift-click and Command-click).*
Many of today's applications support the use of the Shift and Command keys for extended selection. Your part should also support extended selection with embedded parts. If the user holds down the Shift key and clicks anywhere within an unselected frame that is embedded in your active part, select the embedded part. If the part was already selected, deselect the already selected frame. If the user employs the Shift-click combination, you can contiguously extend an existing selection in your active part, if that action is consistent with your selection model.

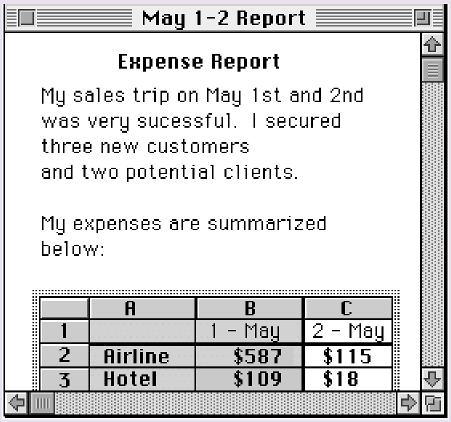
If your part editor supports making a discontiguous selection, including parts that are embedded at the same level in a containing part, your part editor should implement the Command-click sequence.

Command-click is a shortcut for selecting whole parts. Conventional applications sometimes support the use of Shift-click to make a discontiguous selection when contiguous selection is not supported. Your part editor may continue to support this convention, but you should also support Command-click to enhance the predictability of your parts' behavior.

For more information, see pages 563 and 567 of the *OpenDoc Programmer's Guide.*

• *Use background selection.*
A selection in the active window becomes a background selection if the window becomes inactive. If the user presses the mouse button and drags while the mouse pointer is over a background selection, your part editor should perform a drag operation. Thus, one use of background selection is to give users a visual cue of what's selected, allowing them to drag selections from documents in the background.

A Partially Visible Frame in a Root Part



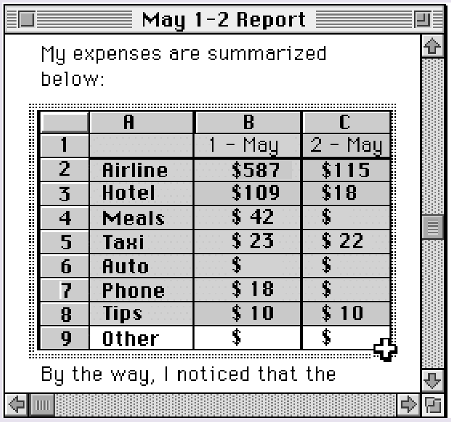
Expense Report

My sales trip on May 1st and 2nd was very sucessful. I secured three new customers and two potential clients.

My expenses are summarized below:

	A	B	C
1		1 - May	2 - May
2	Airline	$587	$115
3	Hotel	$109	$18

Scrolling to Reveal Content



My expenses are summarized below:

	A	B	C
1		1 - May	2 - May
2	Airline	$587	$115
3	Hotel	$109	$18
4	Meals	$ 42	$
5	Taxi	$ 23	$ 22
6	Auto	$	$
7	Phone	$ 18	$
8	Tips	$ 10	$ 10
9	Other	$	$

By the way, I noticed that the

For more information, see page 200 of the *OpenDoc Programmer's Guide.*

• When a part becomes inactive, clear the selection.
There is only one selection per document. This makes OpenDoc documents behave just like today's applications. When a part deactivates, it is responsible for clearing its selection—an inactive part in an active window should *not* use background selection.

For more information, see page 199 of the *OpenDoc Programmer's Guide.*

• Support autoscroll.
Suppose the user clicks inside a partially visible embedded frame and then drags to select some content that is not currently visible. In this case, the containing part or the window should scroll until the user stops selecting or until the bottom of the embedded part's frame is visible. The figures "A Partially Visible Frame in a Root Part" and "Scrolling to Reveal Content" show this situation.

Display at least one unit of overlap below the selection to provide the user appropriate context in the document. For example, if the user is scrolling through text, leave at least one line of text below the selection. In a spreadsheet part, one unit is a row or a column, depending on the direction of the scrolling.

For more information, see page 578 of the *OpenDoc Programmer's Guide.*

• If your part supports scroll bars, you should provide a way to hide them.
We generally suggest that your part not display scroll bars when it's embedded. It adds visual clutter to the user's document and complicates scrolling. But if you do need scroll bars (for example, for a database list part), allow the user to turn the scroll bars off—for example, through a setting.

For more information, see page 535 of the *OpenDoc Programmer's Guide.*

• When your part is inactive, clicking a control should activate the part and perform the operation.
The best example of this is if an embedded, inactive part has a scroll bar. Clicking on the scroll bar should activate the part and start scrolling. The user should not have to first activate the part and then click the control.

For more information, see page 535 of the *OpenDoc Programmer's Guide.*

• Your part should support drag modifiers (for example, Option, Control, Command).
When a user drags a part within a document, the default behavior is to move it. If a user drags a part out of the document, the default behavior is to copy it. However, sometimes the user wants to override that behavior. Option-drag always performs a copy operation and Control-drag always performs a move operation. The Command-drag combination displays the Paste As dialog box, which allows users to create links or override default embedding behavior.

For more information, see pages 597 and 604 of the *OpenDoc Programmer's Guide.* ♣

CD HIGHLIGHTS

continued from page 14

Spellswell 7 Demo from Working Software, a HyperCard Word Services XCMD, and the complete source code to Writeswell Jr., a simple Word Services word processor.

Note: The Word Services SDK, including the specification and sample code, are copyrighted by Working Software, but the copyrights given in the code grant permission to distribute it.

WorldScript Updater

This installer will install files needed by WorldScript technology for users running Macintosh System 7.5 and later. Users of the Japanese and Chinese Language Kits or applications that depend on WorldScript technology should install these files, because they improve the stability and performance of your system. QuickDraw GX and Power Macintosh users should upgrade to these files, because they provide important improvements over previous versions. For details, see the document WorldScript Installer Read Me.

Alex Dosher
Developer CD Leader &
Online Content Librarian

AppleDirections

Business

Feature: New Macintosh user data from Apple's June 1996 U.S. installed base study can help you plan future products and prepare for the transition to Mac OS 8. This article includes data on Mac OS version use, CD-ROM drive adoption, Internet use, and more.

SPECIAL MARKET REPORT

The 1996 Macintosh User Profile

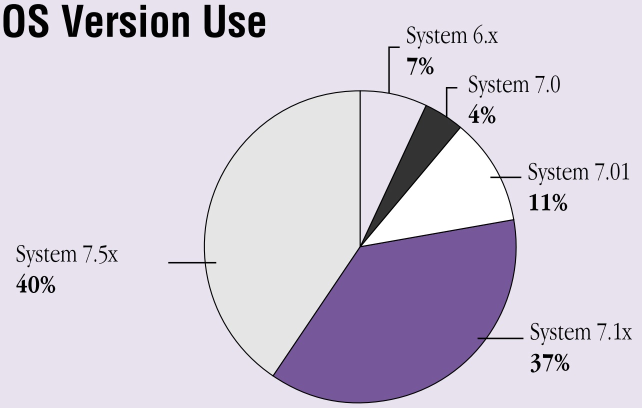
Data to Help You With Product Planning and the Mac OS 8 Transition

Apple Computer, Inc., completed its latest study of the U.S. Macintosh installed base* in June 1996, and in this exclusive *Apple Directions* report, we present you with user data that will help you plan your future products and prepare for the transition to Mac OS 8. Changes to this year's study have made the data more applicable to emerging trends and Apple's new market focus. An enhanced software configuration checker provided more detailed information on system memory and peripherals. Customer usage statistics have been organized according to users' primary job functions, so it will be easier for you to evaluate minimum system requirements for your particular market. And we've broken out information on users with PowerPC processor–based computers—the users who are most likely to upgrade to Mac OS 8—in order to help you write specifications for upcoming Mac OS 8–compatible products.

Since *Apple Directions* published the last user profile (January 1995), two dominant factors have influenced the characteristics of the Macintosh installed base: growing sales of Power Macintosh computers and the popularity of the Internet. These and other factors have contributed to the following installed base findings:

- As of January 1, 1996, 92 percent of U.S. Macintosh customers were running Macintosh System 7.0 or a later version, and 40 percent were using Macintosh System 7.5.
- The CD-ROM drive connection rate more than doubled last year, growing from 19

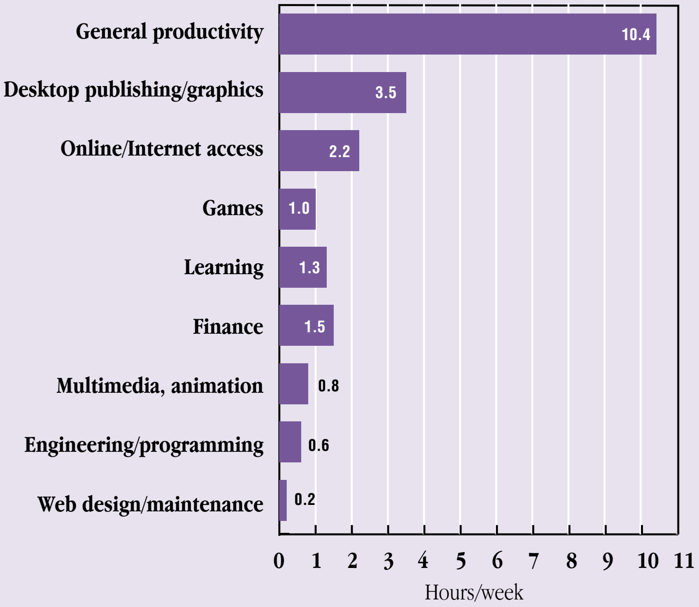
Mac OS Version Use



As of January 1996 nearly 80 percent of the U.S. Macintosh installed base used System 7.1 or higher, as this chart shows.

Source: 1996 Macintosh User Profile Study. © 1996 Apple Computer, Inc. All rights reserved.

How Macintosh Users Spend Their Computing Time



Here are the number of hours per week the "average" U.S. Macintosh user spends on a given computing task.

Source: 1996 Macintosh User Profile Study. © 1996 Apple Computer, Inc. All rights reserved.

*This study is representative of about 86 percent of the active Macintosh installed base. For details, see the box "About this Study" on page 27.

percent in June 1994 to 44 percent in January 1996. Eighty-eight percent of Power Macintosh desktop units were connected to CD-ROM drives.

- Internet use soared from 6 percent to 40 percent between June 1994 and January 1996.
- Twenty-eight percent of Macintosh Internet and online service users spent four hours or more a week online.
- Of Power Macintosh desktop users, 31 percent had access to a color printer as their primary printer and 82 percent had access to high-quality 16-bit audio play and record capabilities, supporting Apple's number one position in the multimedia market.
- Fifty-two percent of Power Macintosh computers were shared by more than one user.

System Software Usage—Seven's Up

Overall, 92 percent of U.S. Macintosh customers were running System 7 or a later

U.S. Macintosh User Configuration Data by Market, January 1996

	Engineering architecture, R&D, mfg.	Graphics, publishing, multimedia	Learning				Medical, health	Home
			Institutions	Students	K–12	Higher education		
	%	%	%	%	%	%	%	%
RAM on board								
4 MB or less	9	14	30	40	30	21	21	33
5–8 MB	37	18	50	30	56	42	32	39
9–12 MB	10	9	8	11	7	10	29	11
16–32 MB	35	33	11	15	6	21	18	15
33 MB or more	9	26	1	4	0	5	1	2
Total storage, all volumes combined								
Less than 200 MB	26	25	54	63	53	50	36	57
200 MB–299 MB	17	21	13	14	15	13	19	19
300 MB–499 MB	9	8	5	5	3	4	18	8
500 MB–699 MB	12	11	17	6	16	13	6	6
700 MB–999 MB	15	4	7	7	7	6	0	5
More than 1 GB	20	30	5	4	5	14	21	5
Screen size (pixels)								
512 x 342 (built in)	5	5	11	14	9	19	9	14
512 x 384 (12-inch)	2	4	10	15	12	4	28	12
640 x 480	41	34	69	55	74	51	48	60
832 x 624	16	11	4	9	2	13	5	5
1024 x 768	6	21	6	1	0	0	0	1
1152 x 864/870	7	12	2	0	0	5	0	1
Maximum pixel depth								
1 bit (2 colors)	16	4	8	4	5	12	4	7
2 bit (4 colors)	5	2	1	3	0	1	0	1
4 bit (16 colors)	3	4	3	7	3	9	11	5
8 bit (256 colors)	38	34	38	44	37	38	36	40
16 bit (65.5K colors)	21	26	43	26	49	22	39	34
24 bit (16M colors)	12	25	2	5	1	10	1	3
Peripherals								
Modem	4	51	29	61	26	43	82	62
CD-ROM drive	34	50	51	46	55	38	52	49
LAN connection	74	49	44	22	45	66	41	12
Audio support								
Internal stereo speakers	61	53	40	50	34	62	56	42
Sound input device present	68	71	89	84	91	90	78	81
16-bit audio play/record	25	31	10	24	5	23	16	16

This chart contains Macintosh configuration data organized in categories that correspond with Apple's core user market segments. The numbers listed in each column represent the percentage of Macintosh computers within a given category that have the specific system attribute or peripheral listed in the left column.

Source: 1996 Macintosh User Profile Study. © 1996 Apple Computer, Inc. All rights reserved.

version and 40 percent were using System 7.5 as of January 1, 1996. This information is not only useful in deciding which operating system versions to support today, but it can also help you estimate Mac OS 8 adoption rates. More specifically, after System 7.5 shipped in September 1994, almost a third of the installed base was running System 7.5 after the first year, and almost a half of the base had System 7.5 installed two years after launch.

About This Study

The 1996 Macintosh User Profile Study was conducted by Evans Research Associates in San Francisco, California, for Apple Computer, Inc. Test subjects for this study were selected from U.S. warranty registration cards for Macintosh systems shipped between January 1, 1990 and January 1, 1996. From this pool of users, 2,600 qualifying Macintosh users were interviewed by phone, then sent a software configuration checker that compiled the configuration details of their Macintosh systems. In calculating overall Macintosh base numbers, collected data was weighted according to computer shipments during the five-year shipment period, in order to more accurately reflect the composition of the installed base. Overall, Apple estimates that this study is representative of 86 percent of the active units in the market place.

CD-ROM Adoption Doubles

One welcome surprise in the study was the number of users with CD-ROM drives. This year the CD-ROM connection rate more than doubled, rising from 19 percent to 44 percent between June 1994 and January 1996. CD-ROM adoption was fairly consistent across all market segments. Zeroing in on Power Macintosh desktop computers, a whopping 88 percent were connected to these devices. This news supports the decision of many high-end application developers to use CD-ROM as their primary application delivery medium. (You've got to respect those shrewd marketers who reaped the savings of delivering products on CD-ROM discs *and* charged users more for these "value-added" discs.)

RAM and Hard Disk Space Rises

With Apple raising the memory levels in base systems (low-end Macintosh Performa computers now ship with 8 MB of RAM and a 700 MB hard disk drive) and more users upgrading to memory-intensive PowerPC native applications, the Macintosh installed base has seen a significant rise in average onboard RAM and hard disk space. At the beginning of this year, about 60 percent of the installed base had 8 MB or more of RAM and 25 percent had 500 MB or more of hard disk space. The largest group of users with less than 4 MB of RAM were those die-hard Macintosh Classic, Macintosh SE, and Macintosh Plus users, who make up 4 to 9 percent of most market segments. (Suffice it to say that these loyal users are not "early adopters" of new technologies.)

Internet Use Increases Six-Fold

The rise in modem use has been steep and steady: By January 1996, 51 percent of Macintosh computer users had modem access, compared with 42 percent in June of 1994 and 28 percent in 1993.

This rise, however, pales in comparison to the growth in Internet use. The number of customers accessing the Internet or online services went from 6 percent at the end of 1994 to about 40 percent in 1995. What's more, these Internet "surfers" are spending *a lot* of time online. If you look at all the time spent in various activities on the Macintosh, online work is the third most time-consuming activity overall (as shown in the bar chart on page 28). And if you look at the data in the pie chart on page 28, you can see that 28 percent of all Internet users spend 4 or more hours a week online.

So what does this Internet data mean to Macintosh developers? We think it means that applications in the Internet arena are going to be hot sellers, and you should seriously consider adding Internet features (Cyberdog functionality, live links to your World Wide Web site, and so on) to your upcoming product releases. Also, since a critical mass of your customers will be "wired" in the coming year, you may want to look at the cost-effectiveness of Internet-based support and upgrades.

Power Macintosh and Vertical Market Segmentation

In this study, data on PowerPC processor–based desktop users was listed separately. This should help you plan your Mac OS 8–compatible products. Though these owners tend to be the high-end users of today (see the chart on page 28), they'll soon be the customers who upgrade to Mac OS 8, and their system configurations will probably be fairly representative of the user configurations you'll see a year from now.

Developers who work with sound and color graphics should take note of the data on color printer access and audio support: Thirty-one percent of Power Macintosh desktop computer users now have access to a color printer as their *primary* printer, and 82 percent have high-quality 16-bit audio play and record capabilities. This support should help reinforce Apple's number one position in the multimedia computer market. (See Industry Watch on page 3 for more on Apple's dominance in this market.)

Another noteworthy data point for developers planning Mac OS 8–compatible features is this: More than 52 percent of Power Macintosh computers are used by more than one user at a given site. This is additional incentive to support the Mac OS 8 feature that lets multiple users create customized workspaces and set individual security levels.

And finally, you may have noticed that we've subdivided Macintosh system configuration data into vertical market segments for the first time this year. Segmentation of the data was determined by an individual's job function, rather than the function of the employee's company, so you'll get more accurate information on which to base product decisions. (For example, the data from a graphic designer who works for a large engineering firm would be placed in the "Graphics" section, not the "Engineering" section.)

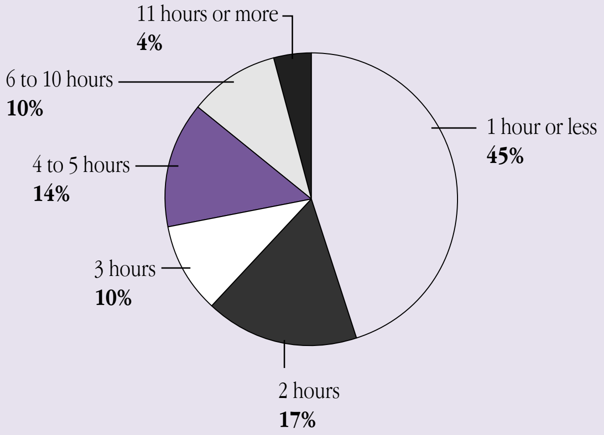
Knowledge Is Power

Though it's nearly impossible to predict anything farther out than a year in the computer industry, hopefully this Macintosh user data will provide you with a concrete basis for product decisions. Looking at this data (through our crystal ball), we think that a year from now a vast majority of your Macintosh customers will have CD-ROM drives and Internet connections. A large percentage will also have rich play-and-record audio capabilities and color printers. Armed with this knowledge and an arsenal of new Mac OS 8 technologies, you'll be ready to create the next generation of powerful Macintosh solutions. ♣

Modem/Internet Use

	All Macintosh systems %	Power Macintosh desktop systems only %
Users with modems		
Modems	51	61
Modem use (% of modem users)		
Have modem speeds of 14.4 Kbps or greater	39	45
Access the Internet or online services	81	87
Use it to send e-mail	78	80
Use it to send faxes	46	54

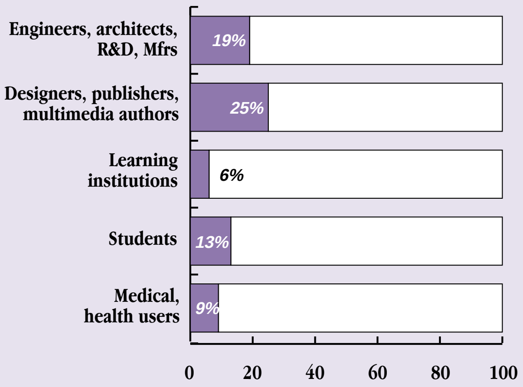
Time Internet/online users spend online



Here's the percentage of U.S. Macintosh systems and Power Macintosh systems that are connected to modems, including information about how customers use their modems, and how much time Macintosh users spend on the Internet.

Source: 1996 Macintosh User Profile Study. © 1996 Apple Computer, Inc. All rights reserved.

Power Macintosh Adoption in Key Markets



This data shows the extent to which Power Macintosh desktop systems have been adopted by key U.S. vertical Macintosh market segments.

Source: 1996 Macintosh User Profile Study. © 1996 Apple Computer, Inc. All rights reserved.

Macintosh vs. Power Macintosh Installed Base

	Total base %	PowerPC-based desktop CPUs %
RAM on board		
4 MB or less	27	5
5–8 MB	39	39
9–12 MB	12	5
16–32 MB	17	59
33 MB or greater	4	17
Hard disk space		
Less than 200 MB	51	3
200 MB–299 MB	17	14
300 MB–499 MB	7	13
500 MB–699 MB	10	28
700 MB–999 MB	5	43
More than 1 GB	9	
Peripherals		
CD-ROM drive	44	88
LAN connection	33	47
Digital cameras	4	7
Video camera	2	4
B&W printer	75	67
Color printer	17	25
Audio support		
Internal stereo speakers	47	100
Sound input device present	81	98
16-bit audio play/record	18	82

This data shows the marked differences between configurations of Power Macintosh desktop systems when compared with the entire base of Macintosh systems in use in the U.S.

Source: 1996 Macintosh User Profile Study. © 1996 Apple Computer, Inc. All rights reserved.

AppleDirections

Listings

IT SHIPPED!

The following 82 products were added to the It Shipped! database between May 15 and June 15, 1996, bringing the total number of products entered into the database to more than 560 since Apple reinstated the program in late 1995. Products developed first for the Mac OS platform are designated with the words “Mac First” on the far right; products available only for Mac OS systems receive the designation “Mac Only.”

The It Shipped! database is used by Apple employees when they prepare advertising, collateral, and white papers and when they help customers find computing solutions; it’s also broadcast to key industry publications.

For more information about the It Shipped! program, visit the It Shipped! Web Page (http://dev.info.apple.com/itshipped.html).

To enter your Macintosh product in the database, you can obtain a form from the site http://dev.info.apple.com/thirdparty/submission.html.

You must also send a copy of the product to Apple at this address:

Apple Computer, 1 Infinite Loop, M/S: 301-1ES, Cupertino, CA 95014, USA.

Product	Company	Mac Only/Mac First
AirWorks	TuneBuilder	
Aladdin Systems, Inc.	Aladdin Desktop Tools	Mac Only
Aladdin Systems, Inc.	CyberFinder	Mac First
Aladdin Systems, Inc.	DropStuff with Expander Enhancer	Mac First
Aladdin Systems, Inc.	SITcomm	
Aladdin Systems, Inc.	StuffIt Deluxe	Mac First
Aladdin Systems, Inc.	StuffIt Expander	Mac First
Ambrosia Software, Inc.	Escape Velocity	
Atlantic Associates	ClassManager	Mac First
Atlantic Associates	QuoteManager	
Atlantic Associates	SeminarManager	
Atlantic Associates	ServiceManager	
Atlantic Associates	TimeManager	
Atlantic Associates	TimeShare Manager	
ATTO Technology, Inc.	AccelNet	Mac First
B.C. Software, Inc.	Final Draft	Mac First
Casady & Greene	Let’s Keep It Simple	Mac Only
Chena Software, Inc.	InfoDepot 2.5	Mac Only
Conceptual Design	EightyRez	
DataCraft	DataWave	
Dayna Communications	BlueStreak 10/100 LC Card	
Dayna Communications	BlueStreak Fast EtherPrint 10/100	Mac First
Design Workshop	dw-2000	Mac First
International Transware	EtherWay MP	
International Transware	InterTalk MP	
Interstudio	Digicad 3D 3.5	Mac Only
Interstudio S.R.L.	Geo-Tec B	
KACHI!SOFT	KACHI!Contact Tracker	Mac First
Kernel Productions, Inc.	ChoiceStick joystick adapter	Mac Only
LIEN Labo d’Instrumentation Electronique	OctavePPC 1.8	Mac Only
Logistiques, Inc.	Trilogy Office	Mac First
Maxum Development	PageSentry	Mac Only

Product	Company	Mac Only/Mac First
Maxum Development	RushHour	Mac Only
Maxum Development	WebLock	Mac Only
Mercury Software, Inc.	Mercury Gateway to AT&T Mail	
Mercury Software, Inc.	Mercury Gateway to Envoy	
Mercury Software, Inc.	Mercury Gateway to MCI Mail	
MindSpring Enterprises	MindSpring Macintosh Internet Starter Kit	
MindVision Software	Installer VISE	
MindVision Software	Updater VISE	
MXM Designs	MacAuditor	
National Instruments	VME-NB2040	
National Instruments	VME-PCI8040 Interface Kit	
National Instruments	VXI-NB2040 Interface Kit	
National Instruments	VXI-PCI8040	
Neon Software, Inc.	CyberGauge	Mac Only
Neurodata, Inc.	Digital Electroencephalography (EEG)	Mac Only
Nirvana Research	MicMac	Mac First
Nirvana Research	MicMac Lite	Mac First
Nisseb softwares	Calles Eudora Plugins	
Pakistan Data Management Services	Urdu Computer Nizam	
Pautex JF	Abaccus	Mac Only
Pautex JF	Calculdepi	Mac Only
Pautex JF	Harmonique	
Pedagoguery Software	Gif.glf.giF	Mac First
Positron	MeshPaint 3D	
Power R, Inc.	Model AV-10 Professional Audio Visual Adapter	
Power R, Inc.	Presenter 575	
Power R, Inc.	Presenter 580/5200	
PowerProduction Software	WebBurst	
RST Systems Technology	TreasureMapper	
Sedona Systems	REPRINT	
Shana Corporation	Informed Filler for Newton	Mac Only/Mac First
Single Source	MacAssist Plus	Mac First

Product	Company	Mac Only/Mac First
SiWare	SoftSlide	
Syndetics Research	Synema* Director	
The Beale Street Group	ExposurePro	
The Beale Street Group	GameCheater	
The Beale Street Group	ScreenShot	
Theatrix Interactive, Inc.	Hollywood	Mac First
Theatrix Interactive, Inc.	Juliard Music Adventure	Mac First
TMC-The Mate Company	SCSI SNIFFER	Mac First
To Win Technology, Inc.	AVIS	Mac First
Water's Edge Software	SuperCDEFs	
Water's Edge Software	Tools Plus	
Waterloo Maple, Inc.	Theorist 2.0	Mac First
WEST Ltd.	WEST Desktop Education Server	Mac First
William K. Bradford	Farm Buddies	
WillStein Software, Inc.	GC WorkMate	
Wisdom Quest Multimedia	eLibrary 96 CD-ROM	
XLR8, Inc.	Warp Factor 135	Mac First

DEVELOPER UNIVERSITY SCHEDULE

Developer University (DU) offers a broad range of Mac OS and Newton programming instruction through hands-on classes and self-paced training products. Classes are offered in Cupertino and through selected third-party trainers. The following is a list of upcoming DU course offerings, including when and where they're offered and how much they cost.

Apple Events/AppleScript Programming
5 days/$1,000
Classroom, Self-Paced
August 12–16, Cupertino

Creating OpenDoc Parts
5 days/$1,500
Classroom
August 12–16, Cupertino
September 16–20, Cupertino

Intermediate Programming: 7.5 Topics
Self-Paced

Introduction to PowerPC
Online

Introduction to PowerTalk
Online

Introduction to RISC Technology
Online

Macintosh Debugging Strategies & Techniques
3 days/$900
Classroom

Multimedia Development with QuickTime VR
3 days/$900
Classroom
August 13–15, Cupertino
September 17–19, Cupertino

Newton Programming: Essentials 2.0
5 days/$1,500
Classroom
September 9–13, Cupertino

Programmer's Introduction to PowerPC
Online

Programmer's Introduction to RISC and PowerPC
Self-Paced, Online

Programming with MacApp
On demand—call DU Registrar for more information

Programming with QuickDraw GX
4 days/$1200
On demand—call DU Registrar for more information

Programming with QuickDraw 3D
3 days/$900
Classroom
September 9–11, Cupertino

QuickStart Mac OS Programming
5 days/$1,500
Classroom
August 19–23, Cupertino
September 30–October 4, Cupertino

Scripting with AppleScript
2 days/$600
Classroom
September 9–10, Cupertino

To register for a class or to get a complete course description by fax, call the Developer University Registrar at 408-974-4897.

Course descriptions can also be found electronically at the following locations:

AppleLink—Developer Support:Developer Services:Apple Developer Services:Developer Information:Developer University

Internet—**http://dev.info.apple.com/du.html**

America Online—Computing:Computing Forums:Development:Mac Development Q&A:Developer University. ♣

APPLE INTERNET PAGE

[Editor's note: We're making a change with this month's feature as the result of one reader's very thoughtful suggestion. Jim Taylor of Seattle's Videodiscovery, Inc., pointed out to us that the Internet Page's five pages of links were "totally inappropriate" for paper, and he's right. A paper version of this feature isn't useful to those of you who aren't on the Internet yet. Nor is it of much use to Internet users who can simply go online and find a genuinely useful electronic version of it. So, starting this month, we're going to save a few pages, and a few trees (as Jim suggested), by providing only the new listings and a list of sites. Those of you who use the Internet can go to the Apple Directions *Web page at* ***http://dev.info.apple.com/appledirections/adtoc.html*** *for the entire list, with the site descriptions, the site addresses, and the links. Thanks for the idea, Jim, and we hope the rest of you like this change.]*

This feature lists Internet resources for online information about Apple Computer, Inc.; its products, technologies, and programs; Mac OS and Newton programming; and other subjects that pertain to the business of computer product development. It includes Internet resources from Apple Computer, as well as from other companies and people. Here, we list the names of the resources, and we describe new sites that we've discovered in the past month. At the *Apple Directions* Web page (**http://dev.info.apple.com/appledirections/adtoc.html**), you'll find descriptions of all the sites listed, as well as links to the sites themselves. We'll update this feature every month, based both on what Apple is doing on the Internet and on your feedback.

New This Month

The following sites are new to the Apple Internet Page this month. Sites provided by Apple Computer, Inc., are marked with the symbol Ⓐ. Although we can't guarantee the content of non-Apple sites, we think you'll find them useful. Know of a particularly useful Internet resource for Apple platform developers? Whether it's a Web page, a list server, an FTP site, or a newsgroup, let us know about it and we'll consider adding it to this feature next month. Send your suggestions to a.directions@applelink.apple.com.

Apple Asia Ⓐ

http://www.asia.apple.com

A one-stop source for information about Apple Asia, its product offerings, customers, developers, and partners. You can find market data about specific countries, as well as information about Apple support programs in Asia, including the Apple Asia Developer Program.

Gif.gIf.giF

http://www.cafe.net/peda/ggg/

Gif.gIf.giF is a program for producing GIF animations that's especially useful for creating animated demonstrations of your software that can be placed on the Web.

International Software Development Guidelines (unofficial)

http://www.tiziana.com/check.html

This, the home page for TGP Consulting, offers a useful list of guidelines for developing international software.

Internet Service Providers Directory

http://www.vni.net/thedirectory

A vast listing of Internet service providers, most of them in the United States but many in other locations, as well. This is the only site we're aware of that's also available in Pig Latin (http://www.vni.net/thedirectory/piglatin.htm).

Macworld Expo

http://www.mha.com/macworldexpo/

A preview of the upcoming Macworld Expo to be held in Boston August 7–10, 1996; it provides show highlights, a list of exhibitors, information about Developer Central, and online registration. The site also provides preliminary information about Macworld San Francisco, scheduled for January 1997.

Mercury Software

http://www.mercurysw.com

Includes free PowerTalk gateways for AT&T Mail, MCI Mail, and Envoy.

QKS

http://www.qks.com/

The home page for QKS, developer of SmalltalkAgents, a dynamic, object-oriented, rapid application development environment. Registered SmalltalkAgents developers can download SmalltalkAgents 2.1 at this site.

Software Publishers Association

http://www.spa.org

News from the Software Publishers Association (SPA) about its activities, its software market research, educational programs, and publications. Also includes membership information, if you'd like to join.

TipWorld

http://www.tipworld.com

Go to this site to sign up for an e-mail newsletter that provides daily tips for using Macintosh computers. Newsletters about other computer industry topics—including Windows 95, shareware, industry gossip, and bugs and fixes—are also available through this site.

Internet Resources for Apple Platform Developers*

Ambrosia Cafe
Apple CEO Gilbert F. Amelio
Apple Competitive Information
Apple Computer
Apple Developer Catalog
Apple Developer Relations Key Contacts
Apple Developer Services and Products
Apple Developers Listing
Apple Directions Express List Server
Apple Education
Apple Europe
Apple Forever
Apple FTP Sites
Apple International Developer Services and Products
Apple Internet Servers

Apple List Servers
Apple Media Program
Apple Pacific
Apple Software Licensing
Apple Solution Professionals Network (ASPN)
Apple Strategy
AppleTalk Network System Support
Apple Tech Info Library
"Ask Apple" Tech Support FAQs
Brad's WebSTAR/MacHTTP
CI Labs
Complete Conflict Compendium
ComputerWare's Macintosh Links
Cult of Macintosh
Cyberdog
DayStar Digital
Developer Depot Catalog
Development Tools
Digitool (Macintosh Common Lisp)
Electronic Publishing Risks
Firewire (IEEE 1394)
Gradient—DCE for the Macintosh
guideWorks
Guy Kawasaki's EvangeList List Server
Guy Kawasaki's Semper.fi List Server
Happy Puppy's Macintosh Games Page
Hartsook Letter
IBM Microelectronics
It Shipped!
Linux for Macintosh (MkLinux)
Mac*Chat Newsletter
MacHack
Macintosh Advantage
Macintosh Application Environment
Macintosh Help Wanted
MacintoshOS.com
Macintosh PowerBook and Mobile Computing
Macintosh Prices, France and United States
Macintosh Programming Tools
Macintosh Speech Recognition
Macintosh Vendor Directory
Mac OS
Mac OS 8
Mac QC Links
MacTech Magazine
Mac vs. UNIX Web Server Performance
MacSciTech
Metrowerks
Motorola Computer Group
Nathan's Everything Macintosh Page
Natural Intelligence
Newton
Nisus Software
OpenDoc
OpenDoc List Server
OpenDoc Part Ideas
Open Transport
Part Merchant Home Page
PC Fairy Tales
Pictorius
Pippin
Polymorphic E-zine
Power Macintosh
QuickDraw GX
QuickDraw GX Fan Club
QuickDraw 3D
QuickTime
QuickTime Live!
QuickTime VR
Quinn's Human Interface Subtleties
Shareware.com
Software Unboxed
Speech Technologies
Technotes
Third-Party Products
Ultimate Macintosh Page
User Group Connection
Web Promotion Services
Yahoo

* Locations (URLs) for these resources can be found at **http://dev.info.apple.com/appledirections/adtoc.html**

Internet Resources for This Issue

News

- Main Apple Web site—http://www.apple.com
- Apple Developer World page—http://devworld.apple.com/devworld.shtml
- Developer University Web site—http://wwwdev.info.apple.com/du.html
- Newton developer home page—http://dev.info.apple.com/newton
- Natural Intelligence Web site—http://www.natural.com
- Digital Harbor Web site—http://www.dharbor.com
- Adrénaline Software Web site—http://www.adrenaline.ca/
- WorldSoft Web site—http://www.worldsoft.com/
- 6prime Web site—http://www.6prime.com
- July 1996 Strategy Mosaic—http://dev.info.apple.com/appledirections/july96/stratmos.html
- PartMaker Pro on the Web—http://www.opendoc.apple.com/dr-live/OpenDocDevelopment/byName.html
- *ODF Developer's Guide*—http://www.opendoc.apple.com/dr-live/index.html
- OpenDoc Web page—http://www.opendoc.apple.com/
- MacApp Web site—http://www.devtools.apple.com/macapp/
- DVD Web site—http://www1.usa1.com/~philk/dvd/index.html
- Katz Media Web site—http://www.Katz.no

Technology

- Apple Game Sprockets—http://devlinfo.apple.com/games
- Apple IC platform press release (May 13, 1996)—http://product.info.apple.com/pr/may.html
- Motorola MPC823 and MPC813 microprocessor for imaging peripherals—http://www.mot.com/SPS/PowerPC/library/press_release/823.html
- List of digital camera vendors—http://www.cgw.com/cgw/vendors/hardware10.html
- TWAIN industry standard protocols for scanning devices—http://www.twain.org/www/whtpaper.html
- QuickMedia Labs Web site—http://www.quickmedia.com
- PictureWorks Web site—http://www.pixworks.com
- Storm Software Web site—http://www.easyphoto.com/storm

Apple Developer Catalog Ordering Information

To place an Apple Developer Catalog order from within the United States, contact Apple Developer Catalog at 800-282-2732; in Canada, call 800-637-0029. For those who need to call the U.S. office from abroad, the number is 716-871-6555. Or, send e-mail to APDA@applelink.apple.com. The Apple Developer Catalog is also available online on the Web (**http://www.devcatalog.apple.com/**).