

AppleDirections

CONTENTS

Increase Your Software Sales With Software Unboxed	1
Apple's Internet Strategy and You Helping You Reach Customers, Today	2
New Opportunities for Component Software Distribution	11
Why I Prefer Macintosh	12
Macintosh Educational Installed Base Gains Five Points This Year, Says QED	13
Macintosh Internet Developers' Association Formed	13
STV Preview Stations Increase Retail Sales	13
Free Radio Air Time	14
Radio (and Web) Promotional	14
CI Labs Releases Live Objects Validation Kit for Mac OS	14
Netscape Navigator for Cyberdog Benefits OpenDoc Developers	14
Apple Dylan for Power Macintosh Ships	14
CD Highlights: System Software/ SDK Edition, October 1996	15
Tales From the Mailbox	16
Multiprocessing Comes to the Mac OS	18
Fear and Coding in Dearborn	20
Creating Packaging That Sells	24
Apple Expands and Organizes Developer Comarketing Programs	28

APPLE NEWS

Increase Your Software Sales With Software Unboxed

If you're looking for ways to broaden your sales and marketing effort (and who isn't?), Software Unboxed gives you two ways to sell your software online. With the first method, they'll do all the work for you, placing your software into a secure digital "shipping container" and posting it on their Web site and America Online site. They handle all Web, phone, fax, mail, and e-mail orders, process credit card payments, and register all new customers. At month's end, you receive a check, a monthly activity statement, and a database of complete registration information for all your new users. Bill Modesitt of Maui Software, which recently offered its TimeSlice product through Software Unboxed, said about this service, "Once I learned of their new system, I immediately joined, and sales have already increased."

With Software Unboxed's second service, you can use their BroadCast technology to sell and deliver your products via the internet, commercial online services, and CD-ROM while maintaining complete control of all receivables and accounting.

Current Mac OS products offered at the Software Unboxed Web site include the following:

- Hungry Frog Series from Science Education Software.
- BoxExpert from Ng Nam Yian
- Outlines in Clinical MedicineAE from K2 Consultants
- MacWise from Carnation Software

please turn to page 11

STRATEGY MOSAIC

Apple's Internet Strategy and You

OpenDoc and Intranets Are Keys to the Future

By Paul Dreyfus, Apple Directions staff

When you closely examine Apple Computer's Internet strategy to determine how your business can benefit from it, you'll find a wealth of opportunities. Many are new, opened by the recent innovations in Internet technology; others involve previously obscure technologies and ideas that have become interesting—even exciting—because of the Internet surge. You'll also find that those opportunities will expand greatly over the coming years, as Internet use and demand for Internet hardware and software products grows.

I've spent some time thinking about the strategy and talking to those at Apple Computer Inc., responsible for carrying it out. To plan how you're going to work within Apple's Internet strategy, I believe the most important points to keep in mind can be boiled down to two words: *intranet* and *OpenDoc*. That's because of the following facts:

- Intranets are the fastest growing part of the Internet, by far. Intranets are private-use networks based on Internet technology that sit behind corporate "fire walls" or in schools or other institutions. They're quickly taking the place of current private local and wide area networks. According to Zona Research, revenues from Internet software will grow approximately 30 percent annually between now and 1998. By contrast, sales of intranet software will grow as much as 100 percent annually over the next three years, and will surpass Internet software sales by almost 3:1 by 1998. Apple is taking those numbers very seriously, placing much of the emphasis of its Internet strategy on intranets

please turn to page 3

For the latest business news, see Apple Developer News at <http://www.devworld.apple.com>; to have the news sent to you, send e-mail to adirections@thing1.info.apple.com. In the subject field, type the string "subscribe <your real name>."

AppleDirections

Volume 4, Number 10

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 (dreyfusp@apple.com)

Technical Editor

Gregg Williams (greggw@applelink.apple.com)

Business & Marketing Editor

Kris Newby (newby.k@applelink.apple.com)

Associate Editor

Anne Szabla (szabla@applelink.apple.com)

Production Editor/Graphic Designer

Lisa Ferdinandsen (lisaferd@applelink.apple.com)

Contributors

Peter Bickford, Valorie Cook Carpenter, Alex Doshier, Dave Johnson

Production Manager

Diane Wilcox

Prep and Print

Consolidated Publications, Inc., Sunnyvale, CA

© 1996 Apple Computer, Inc., 1 Infinite Loop, Cupertino, CA 95014, 408-996-1010. All rights reserved.

Apple, the Apple logo, APDA, AppleLink, AppleScript, AppleShare, AppleTalk, ColorSync, Firewire, ImageWriter, LaserWriter, Mac, Macintosh, MacTCP, MessagePad, MPW, Newton, OpenDoc, Performa, Pippin, PlainTalk, PowerBook, Power Macintosh, QuickTime, and WorldScript are trademarks of Apple Computer, Inc., registered in the U.S. and other countries. AppleGlut, AppleNet, Cyberdog, develop, Finder, Macintosh PC Exchange, QuickDraw, ResEdit, and StartingLine are trademarks of Apple Computer, Inc. Adobe, Acrobat, PageMill, Photoshop, and PostScript are trademarks of Adobe Systems Incorporated or its subsidiaries and may be registered in certain jurisdictions. AppleWorks is a registered trademark of Apple Computer, Inc., licensed to Claris Corporation. DECnet is a trademark of Digital Equipment Corporation. Java and other Java-based names are trademarks of Sun Microsystems, Inc., and refer to Sun's Java-based technologies. Netscape Navigator is a trademark of Netscape Communications Corporation. NuBus is a trademark of Texas Instruments. PowerPC is a trademark of International Business Machines Corporation, used under license therefrom. SOMobjects is a licensed trademark of IBM Corporation. UNIX is a registered trademark of Novell, Inc. in the United States and other countries, licensed exclusively through X/Open Company, Ltd. All other trademarks are the property of their respective owners.

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

Helping You Reach Customers, Today

We're getting behind Apple's commitment to give developers marketing support, in a big way. Starting with this issue, you'll see a great deal more coverage of marketing opportunities opened by Apple Computer, Inc., and other companies—opportunities you can act on today.

Our goal is to help make you as successful in the market as you are in developing great Mac OS, Newton, and Pippin products. Apple Developer Relations (ADR) and others here at Apple are making a priority of working with retailers to increase Mac OS software visibility and sale volume. They're also creating new programs to help you market your products. And every week we're hearing about new ways you can sell your products—including advertising, promotional distribution, and specific sales vehicles.

We'll do our best to tell you about all these activities and how you can take advantage of them immediately. See this month's news, for example, with items about Software Unboxed, component software distribution, and promotional opportunities.

Our new direction (and Apple's) will be especially prominent in the Business section, which will focus on articles that we hope will help you reach customers. Be sure to see this month's Business feature, "Apple Expands and Organize Developer Comarketing Programs," for a guide to immediate comarketing opportunities from Apple.

We'll also point out business and specific marketing opportunities wherever we can in the rest of the newsletter. This month's Strategy Mosaic, for example, includes a great many specific business opportunities opened by Apple Internet strategy as well as a sidebar (see page about a few ways you can work with Apple to reach customers for intranet products.

To make time—and room—for our new coverage, a few things have to go. The Listings section, which included the Developer University schedule, new additions to the It Shipped database, and the Apple Internet Page, is no longer part of the newsletter. The same content, however, is still available on the Web, as follows:

- You can find Developer University information at <http://dev.info.apple.com/du.html>.
- For It Shipped, go to <http://dev.info.apple.com/itshipped.html>.

• The Apple Internet Page, which we'll continue to update regularly, will remain at the *Apple Directions* Web site (<http://devworld.apple.com/mkt/adtop.shtml>), where we hope it will remain a useful "front end" to Apple developer information on the Web.

We'll update you every month about new marketing programs and opportunities; many of you want your information more quickly, though. More immediate updates will be available electronically, on the Web and via e-mail, as follows:

- *Developer World Web site*. All the latest marketing news is featured in Apple Developer News, posted each Friday at the Apple Developer World Web site (<http://www.devworld.apple.com>)

- *Apple Developer News electronic bulletin*. You can also have Apple Developer News (formerly Apple Directions Express) sent to you each week; to do so, send e-mail to adirections@thing1.info.apple.com. In the SUBJECT field, type the string "subscribe <your real name>."

- *Comarketing Opportunities Web site*. This ADR Web site, at <http://devworld.apple.com/mkt/comarketing.shtml>, is updated on almost a weekly basis with new opportunities.

Bookmark these Web sites (and visit them regularly) and subscribe to Apple Developer News to be sure you're getting all the latest and greatest marketing information.

A final note: We'll do our best from our perch at Apple Computer to keep you abreast of marketing opportunities. Please—if you know of any let us know about them, too, so we can share your ideas with the rest of the Apple development community.

Paul Dreyfus
Editor

Internet Strategy

continued from page 1

- OpenDoc lets you give your customers easy access to the Internet, *from within your* application, without their having to exit it and launch a Web browser, mailer, or other special Internet-access software. By turning your application into an OpenDoc container, you enable your customers to add these Internet-access features to your product—features provided by Apple's Cyberdog parts and others' Internet-related OpenDoc parts. In addition, by turning your application into an OpenDoc container you give your customers the ability to use Java™ applets and Netscape™ plug-ins.

In the long run, you'll have the best chance of benefiting from Apple's efforts to pervade the Internet if you adopt OpenDoc and take a very careful look at the market for intranet products. I'll elaborate on these and other opportunities a little later. First, I'll describe Apple's Internet strategy.

A Primer

Please consider what follows a primer on Apple and the Internet; the subject is far too vast to cover in detail in a single *Apple Directions* article. I've tried to summarize the most important points, and steer you in the direction of the opportunities that are currently clearest. We'll return to many of these subjects in future issues to provide more details. Note also that this discussion focuses almost exclusively on the Mac OS, although Apple is also building many Internet capabilities into the Newton and Pippin platforms.

Much of the discussion of the strategy itself is based on a presentation made at the Macworld conference in June by Larry Tesler, vice president of AppleNet, the division of Apple responsible for Internet development. The entire presentation is available on the Web; you can find it at <http://theplan.apple.com/teslerinternet.html>. Apple also maintains a Web site dedicated to communicating about its Internet strategy, technologies, and products; you can find it at

<http://www2.apple.com/internet/>.

The Internet Strategy

Apple's goal is to make using the Internet as simple as using a Macintosh computer has always been. Just as Apple couldn't have made the Macintosh experience what it was without your software, Apple will be turning to you for help in fulfilling its vision for the Internet. Apple will be taking the Mac OS tradition of innovative software design and applying it to the Internet so that previously arduous tasks become easy and the entire experience of using the Internet is improved. Apple's goals are to expand the services provided by the Internet and to make those services available to any user, anywhere.

One aspect of Apple's work on the Internet represents a significant break with Apple's past. Instead of pushing proprietary technologies, Apple will be contributing to open standards that it hopes will make the Internet better for everyone. Anarchy has been a hallmark of the Internet since its inception; while desktop operating system wars have raged, urging computer users to conform to one operating system or another, the Internet has been an increasingly more visible bastion of individualism and free thinking.

Of course Internet technologies have standardized around certain specifications and protocols, such as TCP/IP for communications, FTP and HTTP for servers, POP and SMTP for mail, and so on. However, these standards have remained open enough to accommodate a variety of approaches and materials.

Recognizing that efforts to further "civilize" the Internet are likely to be a huge waste of time—and also learning from its previous effort with proprietary standards—Apple will work toward open standards and an open business model. Of course, where Apple technologies have distinct advantages, especially in the area of content creation and Web servers, Apple will do everything it can to exploit those advantages. Apple also has a vested interest in making its own customers' Internet experiences better and easier than the experiences of other network users. However, wherever feasible, Apple will be contributing its own technologies to make the

Internet as useful as possible to all Internet users.

Apple will apply its technologies and expertise to virtually every area in which people want to work with the Internet, focusing its efforts on

- access—that is, enabling users to connect and view Internet content with minimal hardware and software setup
- delivery—providing the means for managing and distributing Internet-based information as widely and easily as possible
- creation—making it simple for anyone to develop information for use on the Internet

In all three of these areas, tools from Apple and third-party developers are making the Mac OS a powerful, popular development platform. In the area of access and delivery, tools such as AppleScript and Frontier aid Web server automation and CGI development. Natural Intelligence's Roaster and Discover Java from Metrowerks let you develop Java applets and Java server-side software on the Mac OS, while the OpenDoc Development Framework (ODF) simplifies the task of developing OpenDoc/Cyberdog software modules that make Internet access easier.

In the area of content, a variety of authoring and publishing tools from other developers are enabling the creation of media-rich content on the Mac OS for the Internet. Media creation tools for video, sound, images, music, virtual reality and so on, are making the Mac OS the platform of choice for media creation. Other tools that enable the publication of that media-rich content on the Internet and the authoring of dynamic and interactive Web pages—including Adobe PageMill, Home Page, and WebBurst, to name a very small sample—are similarly making the Mac OS the platform of choice for Web publishers.

The following sections show how Apple's Internet development efforts can be loosely organized into the categories just mentioned. In the section "Internet Development Opportunities" starting on page 5, you'll see how you can take advantage of some of Apple's efforts with your own products.

Access

In announcing the Internet strategy at this year's Worldwide Developers Conference, Apple CEO and President Gil Amelio committed to making all Apple platforms—the Mac OS, Newton, and Pippin—"best-of-class" products for Internet access, content creation, and delivery this year. Currently, all Macintosh products ship with Internet access software, such as the Apple Inte-

November Apple Directions Online

November's *Apple Directions* will be available by October 15 on the Web at <http://dev-world.apple.com>.

net Connection Kit, which contains a variety of industry-leading third-party Internet software. Also, Mac OS systems have for some time shipped with MacTCP, which supports the UNIX[®]-based TCP/IP protocol stack used on the Internet.

The latest release of the Mac OS does this one better by including Open Transport, which enables protocol- and connection-independent networking, so that networking applications written to the Open Transport APIs can work atop any network protocol stack regardless of the type of network connection used to hook the computer to the network.

On the hardware side, Apple supports Internet access by ensuring that its computers ship with adequate memory; the latest Power Macintosh and Performa systems, released in August, all ship with 16 MB of memory. Macintosh systems also ship with high-speed networking capability. This translates into Ethernet on desktop computers, and high-speed modems or ISDN for consumer and mobile systems.

As for Newton, the MessagePad 130 with the Newton 2.0 operating system is already the most mobile, Internet-capable device Apple sells, providing Web browsing, Internet e-mail, and newsgroup access with either wired or wireless connections. Apple expects Newton to become an increasingly important part of its Internet access story as future Newton devices are marketed to customers who need instant access to the Internet when they're away from their desktop systems. The Newton operating system will provide an even more versatile way to use the Internet when a Java-compliant version is made available in the near future.

Recent market data shows just how successful Apple's Internet access solutions have been to date. As we reported in last month's news, 28 percent of Internet users polled this year by the Georgia Institute of Technology access the Internet from a Mac OS computer, up from 20 percent six months earlier. Since that's more than twice the Mac OS computer share of the overall computer market, Macintosh customers are more than twice as likely as other personal computer users to connect to the Internet. At the same time, the percentage of Windows clients fell slightly; 58.6 percent said they used some version of Windows, down from 61.5 percent in the October 1995 survey.

To Apple, Internet access means much

For More Information

If you'd like more information on the technologies, products, and ideas mentioned in this article, you're invited to contact any of the following Apple evangelists. They'll be happy to help you pursue your Internet product ideas. If you're not quite sure whom to contact, take your best guess; the evangelist you reach will direct you to the appropriate person, if necessary.

Internet Evangelists

- Jose Carreon, Internet Technologies Evangelism Manager, e-mail: carreon@apple.com
- Jeff Ganyard, Apple Internet Technologies Evangelist, e-mail: ganyard@apple.com
- Mark Altenberg, Mail, Messaging, and Directories Evangelist, e-mail: mark@apple.com
- Carl DeCordova, World Wide Web Technologies Evangelist, e-mail: carldec@apple.com

OpenDoc and Cyberdog Evangelists

- Mark Thomas, Component Software Evangelism Manager, e-mail: markt@apple.com
- Jim Black, Cyberdog Evangelist, e-mail: black@apple.com
- Shaan Pruden, Component Software Evangelist, e-mail: pruden@apple.com
- Eric Shapiro, Internet Components Evangelist, e-mail: eric@apple.com

more than just hardware with Web-browsing capabilities. In providing Internet access to its customers, Apple is introducing new technologies to give Mac OS users the easiest, most satisfying, most compelling experience possible. As these and other technologies reach the desktop, Apple expects the market to be drawn increasingly to the Mac OS for its advanced Internet access capabilities.

One of these new technologies is Cyberdog, Apple's suite of Internet tools that lets users customize their Internet access and combine Internet functionality with existing applications. OpenDoc, which, as I said earlier, lets you give your customers access to Cyberdog parts and other OpenDoc-based software from within your software, is only starting to find its way into the market, and its proliferation will make a significant difference in Internet access. Project X and the Metacontent Format (MCF)—essentially a new metaphor for navigation of electronic information on the desktop and across the Internet—constitute another technology set designed to give Internet access an entirely new meaning in the near future. (See the section called "Project X and MCF" on page 7 for more on this subject.)

Similarly, Apple is working to incorporate Java into all its platforms. Taking its first tangible step in that direction, Apple recently posted to its Web site a prerelease version of the Mac OS Runtime for Java, which makes it possible to view Java applets from Mac OS applications. In case you've been hiding

somewhere, Java is Sun's flexible programming environment that's become popular as a way of building small software modules, or "applets," that can run with Web pages, regardless of the computer platform used to view the page. Although its use in building Internet applets that bring interactivity to Web pages makes Java a part of Apple's Internet strategy, Java is actually part of a much larger story. For more information, see the box "Making the Mac OS a Great Java Platform" on page 8.

Delivery

To help people deliver information over the Internet, Apple offers a variety of server solutions. The Mac OS-based Workgroup Servers are acknowledged to be the easiest Web servers to set up and administer. Apple's Network Server line provides a UNIX-based (actually, AIX) solution for those who require one and can support it; the "hot-swappable" parts of this product line also provide for ease of management, since customers can replace many parts without having to take the Network Server offline.

Again, data shows that these solutions are meeting with success in the marketplace. Since Apple first started selling hardware servers in 1992, sales have increased 40 percent each year, a growth rate Apple expects to maintain or improve on throughout the decade. The Georgia Tech survey mentioned earlier also indicates a disproportionately high usage of Mac OS-based Web servers;

nearly 25 percent of webmasters surveyed said they used WebStar to set up servers, and 13 percent said they used MacHTTP.

Again, Apple's current and future efforts are intended to boost that success. Under the code name *FutureShare*, the next generation of AppleShare (Apple's file-sharing architecture) is being completely rewritten in native PowerPC code, and it will be based on the TCP/IP protocol stack.

In addition, the Mac OS is the only mainstream desktop operating system that can easily incorporate features that let users set up and maintain personal Web servers from their desktop systems. The first product providing personal Web server services for the Mac OS, ResNova's Web for One, has already been announced; you can find a beta version of it on the Web at

<http://www.resnova.com/webforone/>. Once such services are incorporated into the Mac OS, anyone will be able to create and deliver Internet content from a Mac OS-based computer. When Macintosh customers can easily turn their desktop systems into Web servers, it's conceivable that the number of Mac OS-based Web servers will climb dramatically.

Creation

The Mac OS supports a complete suite of tools from Apple and third parties for creating media-rich, interactive Internet content. These tools, notably QuickTime and QuickTime VR (which Apple refers to collectively as the *QuickTime Media Layer*) and QuickDraw 3D, have made the Mac OS the leading platform for Web content development. A study completed by market researchers at Mirai shows that 41 percent of graphic design for the Internet is done on Mac OS systems, more than any other platform. Also, over 50 percent of video available on the Internet is in QuickTime format.

Apple is also contributing to the future of Web content development. It will expand its tools so that content created by them can be viewed from any platform and Web browser. Additionally, a modified version of Apple's QuickDraw 3D Metafile (3DMF) format will play a large role in creating tomorrow's 3D and animated Web content. 3DMF was selected as the binary file format for VRML (Virtual Reality Mark-Up Language) 2.0—a standard defined by the industry's VRML Architecture Group (VAG)—for building 3D and animation into Web sites. (VRML 2.0 is currently supported by over 50 vendors, including Silicon Graphics,

Apple, Netscape, Sony, and IBM.) The first version of 3DMF for VRML 3.0 can be found on the Web at <http://quickdraw3d.apple.com/>.
Apple and Intranets

As I said earlier, Apple is emphasizing intranets in its work on Internet technologies, looking for ways to penetrate that rapidly growing market. In his speech at the Mactivity conference, AppleNet Vice President Larry Tesler said that approximately one-fourth of Fortune 1000 companies have already deployed intranets in their organizations, and most of the rest are considering doing the same. Apple is predicting that the intranet will be the dominant form for internal corporate networks in the future. Additionally, Apple expects rapid growth in educational intranets, as schools at all levels either jump on the cyber-bandwagon or update their technologies.

The primary reason corporations are moving their networks to Internet technologies is the speed and ease of deployment of Internet solutions. Heterogeneous networks have historically created enormous headaches for information systems (IS) managers as they contemplate communications schemes for making multiple protocol stacks, operating systems, new and old technologies, and applications created under different environments somehow work together; the growth of the Internet changes that.

Internet technologies, based on open standards, have traditionally been designed to interoperate with a variety of systems. As a result, Internet applications can be written once and used throughout the enterprise whether employees use desktop systems running the Mac OS, a flavor of Windows, or UNIX. In that regard, the Internet creates a kind of new platform, because applications written to it will work with the other platforms.

Unlike previous custom corporate networking solutions, current Internet technologies also appear to provide an easier way of incorporating older, so-called legacy systems into modern networks. For example, IBM says it will be using Java to take data on its CICS server system and make it available to anyone with a Java-compliant browser.

The reason Apple expects Mac OS technologies to be adopted for intranets is that, while others debate proprietary technologies for Web servers, browsers, mail systems, and so on, Apple is committed to working with open standards and with any vendor who's

also committed to those standards. Apple is embracing those standards—such as TCP/IP, Java, HTML, VRML 2.0—with its platforms and incorporating them into the Mac OS itself. Apple is also taking some of the key technologies for the Mac OS, such as OpenDoc, QuickTime, 3DMF, and the newly developed MCF, and contributing them as multiplatform standards.

Further, Apple is leveraging some of its previously proprietary technologies by making them available for use on the Internet and intranets. For example, Apple's NetFinder lets webmasters convert existing AppleShare volumes into a Web page, which then retains the familiar Macintosh look and feel. You can find a prerelease version of NetFinder on the Web at <http://cybertech.apple.com/AppleNetFinder.html>.

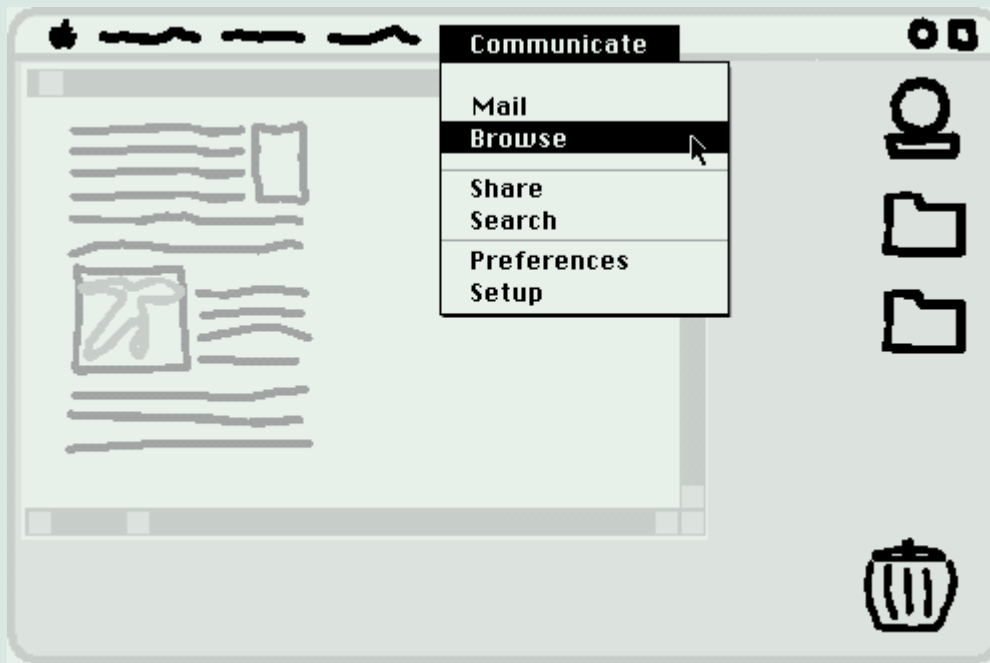
Finally, Apple wants to work with you to define opportunities for providing Mac OS solutions for intranet customers. See the box on page 9, "Intranet Comarketing," for more information.

Internet Development Opportunities

Next I'll describe a variety of specific opportunities for Internet development, as well as future directions you'll want to be aware of. As you consider them, keep in mind that you already have a large and growing group of customers from the ranks of public Internet users. In the near future, you'll also be able to sell to an even larger, faster growing market of people setting up intranets—especially in institutions that are traditional buyers of Macintosh technology, such as publishing and graphic arts companies and educational institutions. Internet hardware and software products are going to have greater potential than you initially thought, because they'll be useful to intranet customers, as well. That CGI you posted to your Web site may be exactly what a major corporate IS manager or an educational system administrator needs.

I'll organize these opportunities and directions under the three areas Apple is emphasizing with its Internet strategy, although some of the opportunities might bridge these categories. Also, just as my presentation of Apple's Internet strategy could only touch on the high points, I can't be all-inclusive with the following sections. There are just too many opportunities to discuss in a single article, and continued Apple engineering and marketing work will uncover significant new opportunities in the near future.

Internet Access From Within Your Application



By making your product into an OpenDoc container, it can give your customers access to Internet features from within the application, as this conceptual drawing shows. All the features shown in the hypothetical Communicate menu bar are delivered by OpenDoc parts.

So, I've focused on presenting the opportunities that are clearest and most actionable at the moment. For more help with Internet products, get in touch with the Apple evangelism contacts in the box on page 4.

Opportunities: Access

First I'll describe opportunities for providing Internet access through Mac OS products.

OpenDoc Container Development. I'll go back to one of the two words I used earlier to reduce Apple's strategy to its essence: *OpenDoc*. In many ways Apple's Internet strategy is simply one more reason for you to adopt this component architecture developed jointly by Apple, IBM, and a variety of other companies, and now endorsed by the more than 2,000 members of CI Labs. (CI Labs is the vendor-neutral organization coordinating and nurturing OpenDoc technology development.)

If you want to give your customers access to the Internet, the best short-term approach is to make your current application into an OpenDoc container. By doing so, you not only make life much easier for your customers by giving them the ability to use the Internet from within your application, but you also give your product potential in the intranet

market as businesses and schools look for the best ways to give users Internet access. You also give your customers access to Java applets and Netscape plug-ins, for free.

Imagine this scenario: Your desktop publishing product (or word processor, database, spreadsheet, or other product), which you've redone as an OpenDoc container application, has a new menu called *Communicate* (or *Internet*). This menu might include the following items: Mail, which lets users send as mail the document they're working on from within that document; Browse, which opens a Web browser inside the document; Share, which immediately posts the document to a part of the user's hard disk that's been set up as a personal Web server; and Search, which provides access to an Internet search engine, again from within the document. (See the schematic drawing above, "Internet Access From Within Your Application," for an idea of what this might look like.)

This approach has another benefit: OpenDoc parts and containers are cross platform and will work under AIX and Windows, giving you a large potential market for your OpenDoc software. IBM is developing OpenDoc for those platforms, as well as for OS/2. You can download beta versions of OpenDoc for AIX

and Windows from the IBM Club OpenDoc Web site (<http://www.software.ibm.com/clubopendoc/tools.html>). Final versions are expected later this year.

Each of the functions in this scenario is delivered by an OpenDoc part, or "Live Object"—the commercial name for OpenDoc software that's been certified by CI Labs. These parts could come from Apple's Cyberdog suite of Internet-access Live Objects, they could be provided by you or others, or they could be in the form of Java applets, which can now be viewed through the Mac OS with an Apple-supplied OpenDoc part that can run Java applets and is part of the Mac OS Runtime for Java. The important point is that by turning your application into an OpenDoc container, you give your customers an extremely convenient way of using Internet services.

Once this kind of seamless Internet access is available, it's likely that demand for it will grow. And it's becoming available quickly: Recently Digital Harbor shipped WAV, a word-processing part that can contain other OpenDoc parts. Corel recently announced that the next Mac OS version of WordPerfect will be an OpenDoc container. Don't you want to deliver the first application in its category/market that's an OpenDoc container

providing Internet access from inside its documents? If your product can provide that kind of seamless Internet access, you can expect it to be of great interest to many customers looking for better ways to access the Internet. If it can't, you can expect your customers to look for other solutions that can.

Live Objects for Internet Access. Next I'll discuss a few opportunities for developing OpenDoc parts that in some way improve Internet access. Cyberdog is Apple's collection of OpenDoc parts that enable computer users to build their own ways of accessing Internet services, including the Web, mail, FTP, newsgroups, and others. Apple invites you to develop your own Internet-service Live Objects, even if they overlap with Cyberdog parts, since you can probably improve on the basic parts Apple has provided in Cyberdog.

To let you know where Cyberdog stands, Apple recently released a beta version of Cyberdog 1.1 (available from the Cyberdog Web site at <http://www.cyberdog.apple.com>). The new version works on 68040-based Macintosh systems as well as Power Macintosh computers. (It will likely run on other 680x0 systems, although Apple has only certified it for use with 68040 processors). Its user interface has also been improved since the 1.0 release. Further, it includes a part for using AppleTalk network system services, and it supports Netscape plug-ins. Finally, Cyberdog 1.1 can be used to view double-byte languages such as Japanese, Chinese, and Korean.

Another improvement made to Cyberdog is that developers can now license Apple's Cyberdog package for inclusion with their Cyberdog parts. This means that if you want to develop a Live Object that works with Cyberdog, you don't have to wait for an installed base of Cyberdog users. You can ship Apple's Cyberdog technologies with your part, and customers will be able to use it immediately.

For a great deal more about Cyberdog, I suggest you take another look at Gregg Williams's article from the February 1996 issue of *Apple Directions*, which you can find on the Web at <http://dev.info.apple.com/appledirections/feb96/opensdoc.html>. However, here are just a few specific ideas for enhancing Cyberdog:

- Parts for working with other protocols not yet supported in Cyberdog—for example,

an IMAP (Interactive Mail Access Protocol) mail part. Internet mail services are currently dictated by two protocols: SMTP (Simple Mail Transfer Protocol), which routes mail from clients to a server, and POP (Post Office Protocol), which then routes mail from the server to the individual clients addressed by the mail. IMAP is a more robust way of routing mail from servers to clients that will likely replace POP over time. While Cyberdog already includes SMTP and POP mail support, an IMAP part has yet to be written.

- Parts for viewing data types that are not yet supported—for example, players for viewing QuickDraw 3D content and for viewing or manipulating future Metacontent Format (MCF) data.

- A version of the Cyberdog browser optimized for certain tasks and markets—for example, a Cyberdog browser optimized for graphics viewing tailored for design professionals.

- A custom curriculum-building part for educators. Such a part would let teachers design documents with content from files on local CDs and Web sites; document content could be customized for each student's needs, interests, and abilities.

- Replacements for current Cyberdog parts that make them more useful for certain sets of customers, as mentioned earlier.

Project X and MCF. Opportunities in this area are particularly interesting, because the technology itself is in a formative state and Apple's engineers want your help in determining the direction it will take. Essentially, they want to be sure they're developing a product that's of greatest use to customers and provides the most attractive developer opportunities.

In its essence, Project X is intended to provide a new metaphor for navigating Internet-based content. It's a 3D viewer that lets users view Internet content that has been defined using the Metacontent Format, or MCF. MCF essentially does for Internet content what the Dewey decimal classification did for library books: It identifies electronic content in such a way that it can be easily organized and found. As the user interface for Project X is currently defined, users appear to "fly" through Web sites and other Internet content to locate the information they need; they can also design their own ways of organizing the content. In fact, MCF can be used to define any electronic data, whether it sits on a desktop or a server. As a result, Project X

can potentially unify desktop and server information into a single viewing environment.

As Project X currently stands, you could provide a number of enhancements, including the following:

- your own viewers for MCF-defined data with user interface elements designed by you
- MCF interpreters
- tools for rendering and clustering MCF data within viewers
- tools for creating MCF data, especially tools that translate current data formats into MCF

You can take a look at a demo version of Project X, as well as a white paper about MCF, at the Apple Research Labs Technology Preview page (<http://www.atg.apple.com/go/projectx/>). Download Project X, play with it, and see what you think. Although documentation is available at the Web site, you need only a few commands to use the demo: To go forward through the information, press and hold the mouse button down; hold the Shift key down at the same time to go faster. When you reach a button with a subject you'd like to explore, release the mouse button momentarily, and then double-click the on-screen button. Press the Command key to move backward, and press Command-T to go back to the top of the demo. (You can send your feedback by e-mail to Apple Internet Technologies Evangelist Jeff Ganyard at ganyard@apple.com.)

Opportunities: Delivery

The next types of opportunities I'll describe are in the area of products that deliver Internet content. These opportunities are primarily for products that work with Apple's Mac OS-based Workgroup Servers, which, as I pointed out earlier, make up more than 20 percent of the current Web server base worldwide. Incidentally, there is also an opportunity to provide products for Apple's AIX-based Network Servers.

Delivery opportunities also include CGIs, or common gateway interfaces—small software modules that sit on Web servers to perform a specific task, usually related to the gathering and presentation of information on the server. Many people have already contributed CGIs, and huge opportunities in this area undoubtedly remain.

For example, content on the Internet is moving from a static, file-based experience, where Web sites provide the same view to

every visitor, to a more dynamic experience involving databases of different types of data that create different viewing experiences. Web sites might recreate themselves “on the fly” depending on the time of day they’re viewed, the kind of browser being used, whether or not a user has privileges for looking at certain types of information, and where a user resides. CGIs for so-called media servers that

organize data in these ways will become increasingly valuable for delivering the future content of the Internet. Additionally, tools for developing CGIs, client and server plug-ins, and dynamic Web pages are a potential long-term opportunity, especially for the increasing ranks of developers without hard-core programming experience who are creating Web client and server software through CGIs

and plug-ins. However, the subject of CGIs is vast enough to warrant a separate article, so I won’t try to do it justice here.

Current Mac OS Web Server Software.

There are a variety of opportunities for enhancing just the existing base of Workgroup Servers. That base can only increase as intranet development takes off

Making the Mac OS a Great Java Platform

Java is far more than a way to liven up a Web page with dynamic content from Java applets, although that’s how Java has gained its early fame: Java is a way to develop applets and, eventually, applications that run on all platforms. For developers—whether you work on commercial products or in-house solutions—that literally makes Java a dream come true. You can write a single version of your code and have it run on every mainstream operating system (and then some).

Developed by Sun, Java is a new, C++-like programming language that many agree is better than C++. The Java virtual machine that runs applications written in Java can be implemented on different operating systems so that Java software will run on each platform, independent of the hardware. Java also includes a set of classes that give programmers easy access to a broad variety of features. Also, Sun is currently working on its own Java operating system and Java-specific microprocessors. In short, Java is quickly becoming a new platform.

Apple wants to make the Mac OS a great platform for running and developing Java software, so Apple engineers are currently hard at work integrating Java into the Mac OS. Additionally, Apple is committed to building Java into its other platforms, Newton and Pippin. Since Pippin is based on Macintosh System 7.5, it will also be able to use Mac OS Runtime for Java technology. Apple expects to announce its plans for implementing Java for Newton soon.

Working closely with Sun on these Java implementation efforts, Apple has licensed the Java virtual machine and run-time environment from Sun. The first fruit of Apple’s labors on this front is the Mac OS Runtime for Java alpha version, which you can find on the Web at the Apple Developer Tools site (<http://www.devtools.apple.com/mrj>). The final version will be available early next year, and it will eventually ship with the new Mac OS.

Mac OS Runtime for Java is designed to let all Mac OS customers run Java software, in whatever way is most convenient to them, whether it’s directly from the Mac OS, through OpenDoc, or through a standard application. The early release includes the following components:

- a Java virtual machine, including byte code interpreter, garbage collector, and verifier
- a standard set of eight core Java classes, such as AWT, lang, net, IO, and others
- a Java applet viewer implemented as a Live Object, allowing a Java applet to be run from Cyberdog and any OpenDoc container
- a stand-alone application for viewing Java applets from within the Mac OS

The third of these components, the Java Live Object, is especially important. If you develop an OpenDoc container, you get Java “for free,”

which means that your customers can run Java applets from with your software as they would any Live Object. This demonstrates Apple’s commitment to integrate OpenDoc and Java; it also gives you yet another reason to adopt OpenDoc.

Also, Mac OS Runtime for Java takes a large step in enabling Java development on the Mac OS. The early release of Mac OS Runtime for Java lets you make your applications “Java-aware” through a high-level API, which means that you can give your customers access to Java applets from within your application. The final release of the product will include the high-level API as well as a lower-level invocation API, which will allow you to load Java classes, create Java objects, and call Java methods in order to build hybrid Mac OS–Java applications. The final release will also include enhancements to make Java software run faster and in a robust fashion from the Mac OS.

By itself, Mac OS Runtime for Java helps turn the Mac OS into a Java development platform, because it will let you write Java software by means of the high- and low-level APIs. Any good development platform also requires tools to make programming faster and more efficient. The good news for the Mac OS is that there are already some excellent tools available for writing Java software on the platform, including Java SDK from Sun Microsystems, JavaScript™ from Netscape Communications, Café and Visual Café from Symantec, Discover Programming With Java and CodeWarrior With Java from Metrowerks, Roaster and Roaster Professional from Natural Intelligence, ActionLine from Interactive Media, WebBurst from PowerProduction Software, and Liquid Reality from Dimension X.

There’s no question about the significance of Java, because it does deliver on the promise of true cross-platform deployment of a single application. In its efforts to embrace new standards, Apple is doing everything it can to integrate Java into its platforms on an ongoing basis, especially focusing on the Mac OS and making it a great platform for running and developing Java software. Also, Apple is going to provide access to some its key technologies—such as QuickTime—to Java developers.

Through these efforts, Apple is not only allowing you to write Java software; it’s encouraging you to do so. We’ll keep you posted about Apple’s continuing efforts with Java.

and there's increased demand for Apple's easy-to-administer Web servers.

- **Utilities.** First, there are opportunities for network developers to market utility software that will make the Workgroup Servers more versatile and useful, both in the public Internet and on private intranets. These include utilities that perform the following functions:

- encryption
- compression
- authentication
- logging, monitoring, and administration

You can write such utilities to the Open Transport APIs, but they must still be modified to work with each separate networking environment/protocol set, including AppleTalk and TCP/IP; IPX, a protocol set being used increasingly for network gaming; Novell Netware; SNA; DECnet™; LAT; and X.25. This can be a royal pain, but that's where the opportunity lies: It's no fun and not too many people want to do it, but providing such utilities within each environment is necessary to ensure that the Workgroup Servers do the best possible job of delivering Internet content.

- **Services.** The actual services provided by servers—file serving, mail, directory services, and so on—sit atop the protocol sets that determine network behavior. Providing these services for the Workgroup Servers also offers a set of opportunities for network developers.

There's a terrific opportunity to provide Internet directory services and solutions, which provide a server-based method of looking up addresses. This is because such services are severely lacking, at least at present. However, with that opportunity comes a great risk, because the industry has yet to coalesce around a standard for delivering directory services. LDPA (Light Directory Address Protocol) appears to be emerging as a standard in that area, so a brave few among you could pave the way for others by developing an LDAP directory server.

Perhaps the greatest need in this area, though, is for enhanced mail services. Although to my knowledge no one has collected data to back this up, the primary task people carry out on the Internet is sending mail. Think of the amount of time you spend browsing the Web versus the time you spend replying to e-mail messages, and you'll see what I mean.

Proprietary systems that work with only a single platform or protocol set have previously generated high-quality mail solutions. Now, with networking moving to the Internet's cross-platform open standards, current mail solutions have difficulty offering equivalent services to all clients. This means there's an opportunity for cross-platform mail solutions that work well in an open standards environment.

Also, a change in mail protocols—the eventual replacement of POP with IMAP, which I mentioned earlier—opens an opportunity for new mail servers that deliver on IMAP's robustness. POP and IMAP are protocols that route mail from the server to individual clients on the network. Unlike POP, IMAP allows management of mail on the server as well as the client and offers the potential for new ways of managing e-mail messages. IMAP servers can include features that filter mail before it's delivered, so only certain types of mail reach the addressed clients. Other features could redirect mail (for

instance, when someone goes on vacation or leaves a company), or organize mail as the server administrator and users see fit.

Unopened mail could also be kept on the server, so that users can determine when they want mail files delivered to their systems.

Another great need is for commerce and security solutions to work with existing Web servers. Many visionaries have talked about the Web becoming the shopping mall of the future. So far, though, the lack of reliably secure, easy-to-use solutions in this area has prevented widespread use of the Internet for buying and selling goods. Apple has been investigating this area, and it's working on making sure that the enabling technologies for such solutions exist. Whoever can deliver a truly secure, viable commerce solution for the Internet will not only contribute to a revolution; he or she will probably make a fortune, as well.

Similarly, given the increasing use of the Web as a mechanism for delivering software,

Intranet Comarketing

Apple now offers several ways of helping you reach intranet customers with your products. These include two brochures promoting Mac OS–based intranet and Internet solutions, “Apple Intranet Guide” and “Macintosh Internet Solutions Guide.” Both list third-party software and, if your product is appropriate for intranet customers, it's possible that Apple will include it the next time the brochures are revised.

Apple has also invested heavily to develop a Web site that's intended to serve as a model for intranet Web site development. It's a site with corporate information about a fictitious business called *Acme Fruit and Nut*; you can find it at <http://www.acmefruit.com>. It actually shows the software used to set up the site in action, as well as listing the software and the companies that created it; since it's revised and updated regularly, it's another place your products might be showcased, if Apple uses them at the site. Also, if you develop intranet sites, you'll probably find this site to be a very useful model. This Web site has also been downloaded to a CD called the *Intranet Companion*.

Additionally, Apple employs a field force of server sales representatives throughout the world. They're responsible for channel development and demand generation for Apple server solutions, which include third-party products. Apple personnel also attend trade shows and events and host seminars about intranets; at these events, third-party products are regularly showcased.

If you think your products are appropriate for inclusion in any of these vehicles, or if you know of any creative ways you can work with Apple to market server solutions to intranets, send e-mail to server.mktg@applelink.apple.com. Please be as specific as you can about your products and ideas.

If you'd like to see the brochures and CD, they're available through Apple's *Starting Line Catalog*, from which you can order by calling 800-373-0877 from the United States; if you're in another location, send e-mail to s.line.order@applelink.apple.com. Here are the order numbers for the products:

- “Apple Intranet Guide”: L01929A
- “Macintosh Internet Solutions Guide”: L01979A
- *Intranet Companion* CD: L01979A

there's a growing need for servers that specialize in online software distribution. Such servers will be particularly useful in delivering the increasing number of Live Objects and Java applets; since they're small and, usually, designed to be revised more often than traditional applications, they're well suited for delivery by electronic means.

AIX-Based Network Servers. There's also a current opportunity regarding Apple's industrial-strength, AIX-based Network Servers. The Apple Event Object Support Library is available on the AIX server, which provides an excellent foundation for developing closely coupled Mac OS clients for the server. Mac OS clients can be created for any type of interaction with the server, such as running an application on the server, accessing data stored on it, and so on.

There's a specific opportunity to develop Mac OS clients that perform system administration on the Network Servers. These servers give up nothing when it comes to performance; however, since they run a version of UNIX, they're more difficult to set up and administer than the Workgroup Servers. Mac OS client software can do that job much more easily, since it can take advantage of the Mac OS interface instead of less intuitive Motif-based interfaces or the traditional UNIX command line to perform server administration. Currently, very few Mac OS clients for the Network Servers have been created to address this need, providing a wide open opportunity. Since this is an area I expect will interest only a few of you, I suggest you contact Leslie Kern, Apple's AIX evangelist, for information on how you can do this; she can be reached by e-mail at kern@apple.com.

A Quick Look at the Future of Mac OS

Servers. As I mentioned earlier, Apple has explored the area of personal Web servers, software that lets any user turn a Macintosh into a Web server and share files with anyone surfing the 'net. The first such product, Web for One from ResNova, is already in beta, and several other personal Web servers are expected in the very near future.

An analogy here is the AppleShare file-sharing feature. That concept started off as "personal AppleShare"—that is, software that allowed any Macintosh user to turn a part of the desktop into a file server on a local area network. Because Mac OS users on networks began to make routine use of file-sharing features when they became available, it's rea-

sonable to expect that once they have the ability to share files over the Internet, they'll begin to do so. Although it's hard to predict the future, it's not out of the question that, once personal Web servers become available, the numbers of Mac OS servers on the Web will climb from the tens of thousands into the hundreds of thousands, perhaps even beyond that.

This will result in a very different world, far more varied, complex, and confusing than even today's Web. This scenario is at least a year or two from the potential for reality, but as personal Web server products take off, it will be worth your while to contemplate ways of helping Internet users keep track of all the new servers and data that will be made available on a daily basis. Some products might help control access to personal Web servers within organizations, providing indexes of available data and routing information that sits on servers; users could visit these larger servers before logging onto the smaller servers that contain the data. Other products might help obtain information from the servers themselves, launching queries across the network that get information from all the personal servers within an organization. If you have ideas along these lines that you'd like to explore, Apple World Wide Web Technologies Evangelist Carl DeCordova (e-mail: carldec@apple.com) would love to work with you.

Opportunities: Creation

The Mac OS is the number one platform for publishing and creating graphics to be used on the Web, which means that there's huge demand for tools to help people create graphics and other material on Macintosh systems for publication on the Internet. If you design such tools, you'll want to consider where the Internet is going, in addition to where it is today. Here are just a few areas to consider:

- **QuickTime and MPEG.** Multimedia content is the wave of both the present and future on the Internet, and any creation tools you build need to support a variety of multimedia data types. As I said earlier, more than half the video content on the Internet today employs QuickTime. Most of the rest is in MPEG. You can support either standard and still allow many users to view video content, because the just-released QuickTime 2.5 supports playing of MPEG video.

- **QuickDraw 3D and 3DMF.** QuickDraw 3D is being used increasingly to create three-

dimensional graphics for the Internet. Also, as mentioned earlier, a modified version of 3DMF is being used as the VRML 2.0 binary file format. Your content creation tools should ideally support QuickDraw 3D data and 3DMF so that they create 3D objects that can be viewed through present and future browsers.

- **MCF.** Apple will be pushing its new Metacontent Format (MCF) for defining Internet data (as described in the section "Project X and MCF" on page 7). Tools that translate current data into MCF will be very useful in Apple's efforts to create standard ways of viewing and organizing Internet data. Further, Apple is urging the development of new content-creation tools that enable the creation and publication of MCF data—for example, HTML editors that allow users to drag and drop MCF files.

More to Come

As I hope you've seen, Apple's plans for pervading the Internet are broad, specific, and actionable. There are many ideas you can act on today to benefit from Apple's strategic push to be, in the words of Larry Tesler, "all over the Internet." If any of the ideas I've mentioned appeal to you, or if you have your own that weren't covered here, I'd strongly advise you to get in touch with the appropriate Apple evangelist listed in the "For More Information" box (page 4). These evangelists will help you link up with the right people and information to develop your idea.

In closing, keep in mind that this is really only the beginning. As Apple's Internet development team continues to organize itself and develop its strategy and tactics, you'll be hearing about a variety of new technologies and products, all of which will open even more opportunities for you. We'll do our best to keep you apprised of them. ♣

Page One News Story

continued from page 1

To find out how you can take advantage of Software Unboxed services and widen your distribution, visit their Web site at <http://www.unboxed.com/submissions.html>, or send e-mail to Fred Sturtevant at freds@broadcastsoft.com.



New Opportunities for Component Software Distribution

Two Macworld participants demonstrated new World Wide Web-based methods for distributing OpenDoc-based Live Object software, as well as other software products. ("Live Objects" is the commercial name you can use for OpenDoc parts that have been validated by CI Labs; see the story "CI Labs Releases Live Objects Validation Kit for Mac OS" on page 14.) The first of these is BuyDirect.Com, which was demonstrated by CNET Direct at Macworld Boston. BuyDirect.Com is a new Web-based software distribution service that will offer customers the opportunity to purchase Macintosh Live Objects and other component software, in addition to traditional software, starting in the fourth calendar quarter of 1996. BuyDirect.Com will list available software at its Web site and handle transaction details, but customers will actually purchase software directly from developers who included their software at the site. BuyDirect.Com can be found on the Web at <http://www.buydirect.com>.

Also at Macworld, Kantara Development showed PartFinder, a Live Object that automatically finds Live Object components at PartBank, Kantara's component software distribution Web site. In addition to software for downloading, PartBank includes news about component software technologies, including OpenDoc and Cyberdog, as well as resources for developers. So far, Kantara doesn't provide an option for selling component software. You can find PartBank at <http://www.partmerchant.com/>.



Macintosh Educational Installed Base Gains Five Points This Year, Says QED

According to the report "1996-97 Technology Purchasing Forecast" from Quality Education Data (QED), the Macintosh computer now accounts for 41 percent of the K-12 education market, up five full share points from last year. Apple II computers account for 20 percent of the installed base, bringing Apple's total share of computers in U.S. K-12 public schools to 61 percent. As a result, the Macintosh platform enjoys strong support from educational developers: According to the Education Products Information Exchange (EPIE), as of early 1996 there were 4,131 Mac OS education titles—more than double the number of education titles for Windows.

According to Field Research Corporation, there are 6.4 million computers in U.S. K-12 public schools today. At Apple's current 61 percent market share, this translates into 2.6 million Macintosh computers and 1.3 million Apple II computers, a total of almost 4 million Apple computers.

The QED report also shows Macintosh as the computer of choice for more than half—55 percent—of all intended school purchases in 1996-97. This is more than seven times the number of computers educators plan to buy from the next most mentioned brands, IBM and Compaq, each with 7 percent. Further, according to QED's Educational Technology Trends, 8th Edition (December 1995), Macintosh is the most popular multimedia platform in schools, with 77 percent of all K-12 school districts that use multimedia computer systems choosing Macintosh systems. Interest in the Internet continues to grow in the education market, with nearly 40 percent of all K-12 schools currently accessing the Internet. Schools with Macintosh computers are almost twice as likely as those with DOS/Windows-based systems to use the Internet. Forty-five percent of all schools

that use only Macintosh computers access the Internet, compared to 27 percent of schools that use DOS/Windows-based PCs, according to Field Research.

Copies of QED reports can be ordered from Quality Education Data at 1700 Lincoln St., Suite 3600, Denver, CO 80231, or by e-mail at qedinfo@qeddata.com.



Macintosh Internet Developers' Association Formed

For support with your Macintosh Internet product development efforts, you can now turn to a new independent industry group. Called the *Macintosh Internet Developers' Association*, or *MIDAS*, it has the following goals:

- devising and refining standards and techniques for Mac OS-based Internet software
- publishing Mac OS-related standards proposals and documents
- coordinating activities with other standards groups such as IETF and W3C
- sustaining and growing a robust market for Macintosh Internet software

Among the first orders of business MIDAS expects to take up are discussions about Web server standards, including the CGI standard, WebSTAR API, and server-side Java. It will also be developing standards for tighter integration between different types of Internet tools, including content creation tools, scripting environments, and browsers. In addition, MIDAS will examine proposals for providing system-level integration of basic Internet client services such as HTTP, FTP, and mail.

Among the founding members of MIDAS—representing some important members of the Mac OS development community—are Akimbo Systems, Aladdin Systems, Apple Computer, BareBones Software, ClearWay Technologies, Cornell University, InterCon Systems, Maxum Development, Microsoft, Netscape Communications Corporation, Purity Software, ResNova Software, Stairways Software, the StarNine

Why I Prefer Macintosh

In which *Apple Directions* readers answer the question “Why do you prefer to develop on/for the Macintosh computer instead of a Windows-based PC?”

“I Can’t Wait To Write Software for Mac OS 8”

From Manoj Patwardhan, Scholastic, Inc.

ManojPat@aol.com

I started using the Macintosh as a desktop computer about seven years ago. Before that I worked mostly on mainframes and UNIX systems. When I started using the Mac, I got excited about writing GUI applications. Ironically, I bought myself a PC just because it was a little cheaper. I found it quite difficult to adjust to a PC having used a Macintosh for some time. I also had some unpleasant experiences trying to configure my system. Dealing with config.sys and autoexec.bat was no joke. I spent a fair amount of money trying to get a compiler I was comfortable with. And, of course, the user interface was not as good as that of the Macintosh.

So eventually I bought myself a Macintosh IIci. It took me ten minutes to set it up, and I have never had a problem with it since. Learning to write GUI applications on the Macintosh using Symantec’s development environment was incredibly easy. Apple’s programming documentation proved to be excellent. I also ended up spending less money for hardware and software than I did for the PC. I have been a keen Macintosh programmer since then.

Programming the Macintosh has been a pleasure for several reasons:

- I enjoy *using* the Macintosh (not just programming it).
- The *Inside Macintosh* books are extremely well written and reflect Apple’s commitment to high quality.
- Excellent development tools are available, such as compilers from Symantec and Metrowerks, resource editors like Resorcerer, and, of course, MacsBug from Apple.
- I find the Mac OS to be technically superior in many ways. For example, the idea that a file should have two forks is a brilliant concept, although it may seem obvious to Mac programmers now.
- The Macintosh ships with a lot of things installed, such as a SCSI port. I use all kinds of peripheral devices (hard disks, CD-ROM drive, SyQuest drive) and installing and removing them is a breeze.

Indeed, I can’t wait to use—and to write software for—Mac OS 8!

“Our Users Like It”

From Patrick Ballin, The Body Shop International

patrick_ballin@bodyshop.co.uk

There are four biggies for me and my colleagues, who develop as an in-house team using high-end tools—GQL, Powerbuilder, Omnis, and various media authoring packages.

- The platform works. We get the occasional irritating ID01 or ID12, but nothing like the rash of GPFs our opposite

numbers in other businesses report, even though they use the same so-called cross-platform tools. And of course, using tools, we port easily to other desktops if we have to.

- The tools support decent interface objects and handle type and graphics properly. The stuff feels good, it looks good, it appeals. We can make it as rich or as Spartan as we like.
- Our users like it. We actually gave them the choice, and they chose the Mac OS.
- You can network easily in a mixed environment. We use AppleTalk, TCP/IP, and IPX, and we don’t even have to think about it at the development level.

“It’s an Electronic Paper Equivalent. It’s a Miracle!”

From Deb Lake, Rascal Software

LakeGroup@aol.com

I started coding about 10 or 11 years ago on a Tandy 64K Color Computer (affectionately remembered by many as “Coco”). As I was also studying to be a graphic designer at the time, I remember wishing for one thing: that I could combine text and graphics on the same screen. But trying to entice and empower the average person into such mundane tasks as balancing a checkbook or logging their expenses on a Tandy 64K just wasn’t happening. I tried my best to fudge it; I think we all did. But the bottom line was that the computer was visually inferior to paper products and paper methods.

Then, of course, along came IBM. I looked at them, but I never bought one. It was neat stuff, but the visual limitations were still there. So, I continued to push the graphical limits of my Coco for a few more years. . . .

And then I read about a computer called *Macintosh*. I couldn’t believe it. It looked like it, it sounded like it, could it be it? An electronic paper equivalent? If it was true, it was a miracle!

And, of course, it was true. It was a miracle. It changed, for the better, my life as a graphic designer and gave me life as a commercial software developer. The Mac gave me the ability to code from a position of choice and creative responsibility. It empowered me to use my creativity to entice and empower the user to expediently accomplish things. And that is all and that is everything. And that is the Macintosh.

• • •

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.

division of Quarterdeck, and Userland Software. For information on how to join MIDAS, see the temporary MIDAS Web site at <http://www.biap.com/midas/>.

You can also join a list server (Internet mailing list) to get regular updates from MIDAS; to do so, send e-mail to midas@biap.com with the string "subscribe" in the subject field.

STV Preview Stations Increase Retail Sales

STV is a company that's trying to do for the software industry what MTV did for the music industry—it's creating a persuasive preview mechanism to help sell titles. Their strategy: They place colorful market-targeted stations in key software retail locations, enabling users to preview software. Developers who place their software on the STV kiosks also benefit from a "detailing" service, where STV representatives regularly maintain and arrange a supply of software boxes at the demo stations.

The benefits of this concept? You gain additional retail shelf space, and you're assured that a reliable product demo is available at every participating store. You also receive reports on kiosk "hits" and stock shortages. STV offers three types of kiosks: "Hot New Games," "Software for the Whole Family," and soon, "Productivity." STV kiosks have been in stores for more than a year now, and, according to STV, some participating software publishers have reported a doubling of in-store sales.

So what does this service cost? Though costs depend on the retail outlet, STV estimates that a developer may pay \$35 to \$50 per month, per store, per product—a cost that's a fraction of what it would take to purchase a retail shelf "end cap." STV currently has software demo kiosks positioned in Fry's Electronics, BestBuy, Borders Books and Music, Computer City, Virgin Megastores, Media Play, Lechmere, and Elek-Tek.

For more information on STV, contact Melinda Moore at 310-337-3058.



Free Radio Air Time

Do your friends say you talk too much about Mac OS computers? Want a chance to get some free product "air time" on a national radio computer talk show and put in a few good words for the Mac OS platform? Thomas Piowow (tjpa@tjpa.com), the "Mac Guy" on the monthly call-in radio program "The Computer Guys," needs more Macintosh callers. Please tune in and call in your Macintosh tips, tricks, success stories, and questions during his show.

Piowow appears on the "Derek McGinty Show," broadcast at 1 P.M. the first Tuesday of every month on the following public radio stations:

- KUAR—Little Rock, AR
- KVOG—Odessa, TX
- WAMU—Washington, DC
- WFAE—Charlotte, NC
- WFSU—Tallahassee, FL
- WHYY—Philadelphia, PA
- WKNA—Memphis, TN
- WKNQ—Dyersberg, TN
- WUOM—Ann Arbor, MI
- WVPE—Elkhart, IL

(If your city isn't on this list, you may want to ask your local public radio station to look into carrying the show.)



Radio (and Web) Promotional Opportunity

Speaking of radio, we recently learned of a unique new program, called "The Mac Show," which bills itself as the "world's only Macintosh-only computer radio show." It gives you yet another vehicle for promoting your Mac OS products. "The Mac Show" launched in St. Louis on February 23, 1996, and, with RealAudio, it's now also broadcast over the Web. The show covers new Mac OS hardware products, software, educational and financial information, and technologies. To find out about promoting your product through the program, or to submit a product for review, send e-mail to Everett Marshall, the host and producer of "The Mac Show," at everett@macshow.com, or call him at 314-

692-8900. Also, visit "The Mac Show" Web site at <http://www.macstore.com/macshow/>.



CI Labs Releases Live Objects Validation Kit for Mac OS

Do you want to be certain that customers will buy your OpenDoc part with confidence? The way to do this is to make sure it passes the part-validation test set up by CI Labs (Component Integration Laboratories), the nonprofit group that oversees the OpenDoc component-software technology. Once you do that, you can use the trademarked name "Live Object" with your part, and customers will know they're getting a part that will work well in the OpenDoc environment. (For more on CI Labs' "Live Objects" announcement, see "CI Labs Announces 'Live Objects' Name" in the September 1996 issue of *Apple Directions*.)

CI Labs recently announced a product, called the *Live Objects Validation Kit version 1.0 for Mac OS*, that allows developers to self-certify OpenDoc parts and register with CI Labs to be eligible to use the "Live Object" name on their OpenDoc parts.

This version of the Live Objects Validation Kit is now available for downloading, free to CI Labs members. (Membership in CI Labs costs \$350, and you must be a member of CI Labs to get the Live Objects Validation Kit.) To join CI Labs, go to their Web site at <http://www.cilabs.org> or send e-mail to webmaster@cilabs.org.



Netscape Navigator for Cyberdog Benefits OpenDoc Developers

Netscape Communications will develop a new version of Netscape Navigator that supports Cyberdog and OpenDoc, according to a recent joint announcement by

Netscape and Apple. To be called *Netscape Navigator for Cyberdog*, the new product will be an OpenDoc component developed specifically for the Apple Cyberdog Internet suite. Apple will distribute Netscape Navigator for Cyberdog with the Mac OS and Mac OS–based computer systems; the component will be the default Cyberdog World Wide Web browser.

The Netscape Navigator for Cyberdog component will benefit OpenDoc developers at the same time it makes Internet access easier for Mac OS customers. If your product has been modified to work as an OpenDoc container application, your customers will be able to access the Internet from within your application through the new component. For example, with Netscape Navigator for Cyberdog and a word processor with OpenDoc container support, such as ClarisWorks or Corel's WordPerfect, a user would be able to view the Internet from the application and incorporate links to Web sites in the documents they create.

For more information about Cyberdog, including a free copy of the latest release (Cyberdog 1.1 beta 3) visit Apple's Cyberdog Web site at <http://cyberdog.apple.com>.



Apple Dylan for Power Macintosh Ships

Apple recently shipped the Apple Dylan Technology Release for the Power Macintosh computer. Apple Dylan is a development environment for the Macintosh platform based on the object-oriented dynamic language (OODL) Dylan. This is a “technology release” because the software is unfinished; it contains a number of bugs, and it won't be supported or updated through Apple's standard technical support processes. The Power Macintosh Apple Dylan Technology Release is hosted on RISC-based Power Macintosh systems, although you can use it to produce code for both 680x0 and Power Macintosh systems. The release is available either as a stand-alone CD (for developers who already have the documentation) or as a complete package, including the CD and six manuals. The CD includes the previous-

ly released 680x0 version of Apple Dylan as well as the new Power Macintosh version. You can obtain the Apple Dylan release through the *Apple Developer Catalog*.



Apple Announces Plans for Windows Version of QuickTime API

Apple Computer, along with Macromedia Inc., the Media 100 group of Data Translation, and Truevision, announced that they'll collaborate on implementing the complete QuickTime application programming interface (API) for Microsoft Windows 95 and Windows NT. Once their work is complete, developers will be able to create and edit QuickTime content on the Windows platform.

With the full feature set of QuickTime API available on Windows, you'll be able to create content on either a Mac OS- or Windows-based system and deliver that content to multiple platforms with only a single development effort. You'll be able to save time and money while delivering QuickTime content in a consistent manner to a larger market. Playback of QuickTime digital media on the Windows platform has been available since 1992.

Apple is jointly developing this new version of QuickTime for Windows with Macromedia, Truevision, and Media 100, who will provide their multiplatform development experience and testing to ensure the highest possible performance from the software.

“The QuickTime development project marks the first step in Apple's plans to move all of its core interactive multimedia technologies—something we call the QuickTime Media Layer—to key industry operating systems such as Windows, OS/2, and UNIX,” said Marco Landi, Apple's chief operating officer.

“Having truly multiplatform technology for both creation and playback is a major win for PC developers everywhere,” said Ellen Hancock, Apple's chief technology officer and executive vice president of Research and Development. “For the first time, developers can build next-generation

software on our open QuickTime middleware and feel confident that they can save time and money while getting their products into customer hands more quickly and efficiently.”

Today, QuickTime for the Mac OS supports multiple data types, including video, sound, graphics, animation, text, music/MIDI, MPEG, and sprite 3D, with the ability to synchronize all the media types to a common time base. With this release, Apple plans to make all these features Win32 compatible. Content publishers of high-quality, high-data rate and full-screen video will be able to use QuickTime to produce broadcast-quality productions on either Mac OS- or Windows-based personal computers.

To ensure compatibility and a timely release, Apple, Macromedia Inc., Media 100, and Truevision have agreed to a joint development schedule, shared specifications, and combined engineering efforts. Apple plans to begin a widespread beta software seeding program in the late fall. At that time, you'll be able to participate in the final testing and compatibility phase of the project. ♣

Technology

CD Highlights: System Software/SDK Edition, October 1996

Features: Multiprocessing Comes to the Mac OS; Fear and Coding in Dearborn

Human Interface: Tales From the Mailbox

CD HIGHLIGHTS

System Software/SDK Edition, October 1996

Once again, it's System Software time; this month's disc features 15 localized versions of the System Software 7.5.3 universal installer, and 13 localized versions of System Software 7.5.3 for the new Power Macintosh 5400. In addition, the Worldwide System Updates folder contains a worldwide version of the System 7.5.3 Revision 2 updater. You should install System 7.5.3 Revision 2 only if you have a Macintosh computer with System 7.5.3 installed or a PowerBook computer with System 7.5.2, PowerBook Enabler 1.2 or later installed; for details, see the document About System 7.5.3 Revision 2.

In addition to updates to the Developer University schedule, the Gestalt selectors list, and the interfaces and libraries, here are this month's new and revised items.

Apple Game Sprockets

Apple Game Sprockets are Apple's solution for developing games on the Macintosh. They include NetSprocket, for network games; InputSprocket, for joysticks and other game devices; DrawSprocket, providing access to page flipping and double buffering in a single API (application programming interface); SoundSprocket, for 3D sound; and RAVE, a low-level layer for 3D games that need hardware acceleration. Note that the current versions are for PowerPC processor-based computers only.

CFM-68K

CFM-68K is a run-time architecture for 680x0-based Macintosh applications and shared libraries. This architecture is modeled after the Power Macintosh run-time architecture, and it uses the CFM (Code Fragment Manager) to provide shared library support for 680x0 Macintosh computers. CFM-68K run-time applications and shared libraries can coexist and run simultaneously with classic 680x0 run-time applications. CFM-

68K version 1.0.4 features mixed-mode bug fixes since the 1.0.1 release.

ColorSync 2.1GM

ColorSync 2.1 is a second-generation color-management system that offers prepress-quality color matching and separations. ColorSync 2.0 offered substantial benefits for developers; ColorSync 2.1 provides a standard architecture and profile format for color matching on the desktop that delivers the solution that developers have asked for—no more decisions about which API and profile format to use, and no more time wasted writing custom routines. The ColorSync 2.0 API provides the most powerful “plumbing” for color management.

ColorSync 2.1 features

- high quality and performance transformation engine
- expanded, compatible architecture
- ICC Profile compatibility
- support for PostScript™ Level 2 devices
- support for more than four color devices
- increased application support

ControlStrip Test

ControlStrip Test is a sample program that shows how to use some of the ControlStrip calls.

IW-Half-Dither

IW-Half-Dither is a new ImageWriter sample driver for QuickDraw GX that shows how to implement color-plane separations as well as how to change the 'rdip' resource “on the fly.”

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.

PAPWorkStation.o version 8.3.4 fixes a problem that affected some Power Macintosh computers with PCI expansion slots printing to fast printers or print servers. Note, however, that PAP.h and PAP.p remain unchanged.

MoofWars

MoofWars is C++ sample code that demonstrates a few techniques for building efficient animation code on Power Macintosh computers. This code takes advantage of DrawSprocket to provide access to page flipping when the video hardware supports it.

The main features of this application are the graphic and tile classes, which are used to draw the background and then composite sprites on top of it. The blitters within these classes are designed to move data as efficiently as possible on PowerPC processor-based computers—that is, whenever possible, each read or write is done on an aligned boundary, and data is moved in the largest-possible chunks.

This version should work on any Power Macintosh. It requires the DrawSprocket shared library, which you can find in the Mac OS SDK and on the Sprockets Web page (<http://dev.info.apple.com/games>).

OpenDoc Development Framework

This folder contains OpenDoc Development Framework (ODF) Release 1, Update 2. For information about changes in this version, see the file ODF Release 1 Update 2.

This release supports OpenDoc 1.0.x for Macintosh only. To do part development for Windows, you must use ODF 1.0d8, which supports both OpenDoc DR1 for Windows and OpenDoc DR2 for Macintosh. (Because the OpenDoc API changed so much since DR1, it became impractical for Apple to continue to support both APIs in the same code base. Apple is still committed to supporting OpenDoc for Windows when that API reaches a level that allows them to do so.)

Since the 1.0d11 release of ODF on the OpenDoc DR4 CD, portions of ODF have moved

please turn to page 23

Tales From the Mailbox

By Peter Bickford

Regular readers of this column know that about a year ago, my wife and I moved into our first house. And, despite my attempts to electrocute myself building a music studio in what used to be the garage, Carolyn and I have been having a wonderful time. Only one cloud has darkened the skies over our happy homestead: Our mail carrier seems to have a problem with, oh, actually *picking up* our mail.

Now it's not as if the mail carrier never visits our house. I'm one of those folks who sends in his warranty cards and subscribes to a dozen different magazines, so I can count on getting at least an inch or two of junk mail every day. But for reasons that defied comprehension, our mail carrier would just dump the new mail on top of anything waiting to go out, then merrily proceed on his route.

When this happens one time, it's an understandable lapse for an overly busy mail carrier. When it happens three times, it's worth looking into as a problem. But when your soon-to-be-late mortgage payment has been sitting in your mailbox for a week straight because some bloody idiot masquerading as a U.S. Postal Carrier can't be bothered to actually *carry mail*, it makes you a little crazy.

With your permission, let me tell you a story about a man, his mailbox, and how he nearly murdered his mail carrier before he realized that what he really had on his hands was a human interface problem.

On the Deep Cultural Importance of Mailbox Flags

The first time we put mail out for pickup and it was still there the next day (underneath the newly delivered mail), my wife and I did what any user of a new system would do: We blamed ourselves. Carolyn and I just looked at each other and concluded, "Gee, we must have forgotten to put the flag up."

For my readers outside the States, I should explain the last comment. Since the dawn of U.S. postal history, a sacred con-

tract has existed between common citizens and their mail carriers: When you've got something that you want the mail carrier to pick up, you raise a little red flag on your mailbox. The mail carrier in turn picks up your mail, then lowers the mailbox flag again.

This two-part interaction is what passes for social contact in many parts of the United States, and it carries with it additional social subtleties. America prides itself on being a classless society, and is notoriously anxious about any relationship in which a job puts a person in the role of serving somebody else. American waiters, for instance, like to form relationships with their customers, so that serving them food feels a bit like doing a favor for a friend. If their new "friends" are rude enough not to acknowledge the waiter's efforts with a smile, a "thanks," and a suitable gratuity, the waiter feels snubbed and wonders who the heck the diners think they are.

By doing your part and raising the mailbox flag for outgoing mail, you're showing the mail carrier you see your relationship as a partnership, not an act of servitude. If you fail to handle your end of the interaction and leave the flag down, mail carriers will often gently remind you of your lapse by "forgetting" to pick up the mail that was clearly in sight when they went to dump in the new mail. Customers desirous of warm relations with their mail carrier (and a lack of "accidentally" mangled mail during the year) would also be wise to adopt the Midwestern tradition of baking something nice for their hardworking mail carrier at Christmas time. It's through such little reciprocal rituals that the fabric of our classless society is preserved and our grand democratic tradition marches on. Recently, certain provocateurs have attempted to ban public employees, including mail carriers, from accepting gifts of any kind. The result is a society in peril, full of misdelivered packages and hungry, embittered postal workers.

Although it's had a long and distinguished history, the mailbox flag is

apparently suffering as an institution. One sign of its weakness is that it proved remarkably unsuccessful as an interface element when used in the Alarm Clock desk accessory. More telling, however, is the fact that the intrepid men and women who deliver in my area seem to regard it simply as a curious ornament that happens to be attached to an awful lot of mailboxes. It seems never to occur to them that the flag might have some special meaning, or that anything other than the wind might raise or lower it on a given mailbox. So whether I'd raised the flag when putting mail out or not, my mail would still be there the next day, buried under the new day's newly delivered junk mail.

Training and Documentation

My uncollected mail problem began to drag on, and it was time to get to the bottom of the mystery of why my mail carrier wouldn't carry mail. I couldn't think of a reason for the mail carrier to have a vendetta against me (other than the dozen or so *MacWarehouse* catalogs that show up in some variation of my name every month). That it wasn't a personal grudge was confirmed when I checked with the previous homeowners and found that they'd had similar problems for the fourteen years they'd lived at this address. They swore, however, that they'd always been nice to the mail carrier, and that they could think of no reason that their mail carrier might have it in for them.

I then spoke with the postmaster at the local post office and he confirmed that yes, the carriers on my route had not been given any special dispensation from picking up outgoing mail. He asked if I'd been putting the flag up. I said I had. He said he'd talk to the mail carrier on my route. This got my mail picked up for two days, then it went back to being left uncollected.

By now, it seemed as if there was only one possible conclusion about the source of the problem: Our mail carrier was an incompetent git whom fortune had blessed with the IQ of a Chia Pet.

The solution, I decided, was training.

Through several frustrated calls to the post office, my wife and I arranged to meet with our mail carrier and his substitute, chat with them about the problem, and even give a short demonstration of how to pull the handle of our mailbox in order to actually open it and access any outgoing mail. The “training” helped immensely, and seemed to more or less solve the problem for several months. Eventually, however, our trained mail carriers moved on to other routes. Replacing them was a legion of substitute carriers, all of whom had the same old problem picking up our outgoing mail.

In desperation, Carolyn and I decided to augment our old mailbox training with documentation of sorts. This began as a series of progressively ruder letters pinned to our mailbox. (The nicest was “Dear Postal Carrier, What the heck does it take to get you to pick up our mail??!”). Eventually, we broke down and attached “help text” to the mailbox itself: “Raised flag means outgoing mail is waiting” and “Pull handle to open.” As can be expected of most help systems, the people who most needed to read it never even bothered to look at it.

User Studies

By now, the whole thing had turned into a grudge match. Sure, I *could* drop my outgoing mail off at Apple, but there was no way I was going to let the U.S. Postal Service deprive me of decent mail service, which I now believed to be one of the most important rights of home ownership.

It was by sheer luck that I stumbled upon the secret to victory. I had stopped home for lunch and was pulling out of my driveway to head back to work when I noticed my mailman heading down the street. This time, instead of wearily waving “Hi” and reminding him to pick up the outgoing mail, I decided to simply watch and see what he did.

The mailman was distracted. Come to think of it, all the mail carriers I’d ever seen on my route were distracted. As they walked from house to house, they were busily sorting mail for the houses ahead in order to finish their rounds on time. In moves that spoke of years of practice and the need for efficiency, my mailman managed to finish sorting out the mail addressed to my house just as he reached the mailbox. Then, without even looking

up, he dunked the mail into the top slot and continued on his way, already beginning to pull up the mail for the next house.

The whole move took less than fifteen seconds, and once again, he had managed to dump new mail on top of the outgoing mail.

I briefly thought of running him down with my car.

Then I suddenly realized what the whole problem was.

A Solution Presents Itself

One of the many things I liked about our new house was that it had a big mailbox. All the previous places I’d lived in had those little banks of locking cubbyholes that serve as apartment mailboxes. It is an apparent point of postal service honor that neither snow, nor rain, nor the constraints of physical mailbox dimension shall prevent these carriers from delivering the mail. Accordingly, even the frailest of carriers is capable of exerting pressure like unto a 20-ton hydraulic press if that’s what’s required to ram a cubic yard of junk mail into a 3-inch-tall mail slot. (Pity any package marked “fragile” or “do not bend.”)

Mercifully, our new mailbox had a bin large enough that it could hold the day’s mail without physical compression techniques being employed. It also came with a convenient top slot for dumping in new mail, so the bin need not even be opened if nothing was going out. Unfortunately, it was this convenient top slot—not found on the simpler mailboxes in the neighborhood—that was causing all the trouble.

The mail carriers in my area are stretched much thinner than the ones I remembered from my youth in the Midwest. Their routes are longer, the houses are more closely spaced, and the volume of mail is much greater than in years past. For the carriers in my area, the priority is to deliver the mail from house to house as speedily as possible, even if that means sorting mail as they walk between houses rather than scanning for mailbox flags. With practice, they can finish their route and get home to their loved ones without having to work overtime.

It began to dawn on me that I was not the only user of my mailbox. Although the mailbox design worked well for me, the

other users of my mailbox—the mail carriers—were being led into habitual user error by it. The mail carriers’ most pressing concern as they approach the mail box is to drop off the mail they have in their hands. The top slot makes it far too easy for them to accomplish this and be on their way without ever seeing the mail in the bin below waiting to go out.

The next day, I tried a new solution based on the idea that the problem was not that my mail carrier was stupid, but that the box was designed wrong. In the jargon of interface geeks, I added a primitive “forcing function” to my mailbox’s user interface. Using heavy tape, I sealed off the top slot from the inside. The idea was to make it necessary to physically open the bin, revealing its contents, before anything new could be added. If the tape wasn’t up to the job, I was ready to make the alteration very permanent by using liberal amounts of epoxy putty. As it turned out, that step was not necessary. From that day forward, my outgoing mail has been collected without fail.

The Moral

In retrospect, I can’t believe it took me a year to figure this one out. My only comfort is that the previous owners didn’t work it out in 14 years, and had to move to an entirely different neighborhood before their mail pickup improved. Geez, I hope they didn’t get the same kind of mailbox.

There were at least three lessons I learned the hard way from this little saga. First, *just as great human interfaces can make novices perform like experts, badly designed systems can make intelligent, skilled people look like idiots*. Second, the fine points of a user’s job can make the difference between a system that works and an utter fiasco.

And, finally, it’s a lot better to sit quietly and watch your users at work for a few minutes, than to spend 25 years to life in prison for running down your mail carrier.

*Till next time,
Doc*

Peter Bickford is a senior scientist in Apple’s Developer Consulting Group. If you have interface questions you’d like to address to him, send e-mail to bickford@apple.com.

Multiprocessing Comes to the Mac OS

By Gregg Williams, Apple Directions staff

You Can Rewrite Your Applications to Take Advantage of Multiple PowerPC Processors

No matter how fast processors get, some people want even more speed. Even if they were willing to wait for the next generation of faster processors—which they aren't—the incremental speed increase of the next-fastest processor wouldn't be enough to satisfy them. But now there's a solution: You can speed up a certain class of applications by designing them to run on a Mac OS-compatible computer that has multiple PowerPC processors.

Apple Computer, Inc., recently announced that its new Apple MP API (multiprocessing application programming interface) has gone Golden Master. This API, which Apple contracted with DayStar Digital to write, gives developers a single hardware-independent, Apple-endorsed API for doing multiprocessing. This means that you can now modify your application once to obtain an MP-savvy application that will run without modification on a variety of multiprocessor Mac OS-compatible computers from a variety of vendors. (Before, when non-PowerPC processors were used to accelerate specific applications, you would have to create a different version of your application to run with each accelerator card.)

Starting now, multiprocessing is a technology that many users are willing to pay for, and the number of multiprocessor Mac OS-compatible computers will increase in the years to come. Any application that is compute-intensive will benefit from having multiple CPUs working on tasks. Applications that will most benefit from the Apple MP API include those in the following areas:

- image processing
- 3D rendering
- digital video
- scientific and technical

The people who use such applications are professionals who are very interested in getting their work done in the least time possible. If your application supports multiprocessing, it will

have a significant advantage over competing applications that don't.

The Apple MP API is important because it brings multiprocessor support from workstation-level computers (which cost more than \$10,000) to the Mac OS desktop (in computers selling for less than \$6,000—for example, the \$5,699 Power Macintosh 9500/180MP). Since owners of certain Mac OS-compatible computers—those with the PowerPC processor on a removable card—can add two-processor multiprocessing for under \$1,500 by using DayStar Digital's nPOWER 360+ upgrade card, multiprocessing power is within many customers' reach—and the entry cost is sure to be lower in the future.

How the Technology Works

Mac OS-compatible computers that are compliant with the Apple multiprocessing specifications have one main processor and one or more attached PowerPC processors. The main processor runs all applications and the Mac OS. Any code running on any processor can spin off a unit of execution called an *MP task*, which the MP kernel automatically moves between processors to balance the computation load. The MP tasks execute preemptively, so that all MP tasks get execution time on a regular basis.

Previously, to do multiprocessing, both the application and the operating system had to support multiprocessing. To make multiprocessing work under System 7.5 (which does not have multiprocessor support built into it), the designers of the Apple multiprocessing architecture had to force one significant limitation onto MP tasks: namely, that MP tasks cannot call, directly or indirectly, any Toolbox or Mac OS routine. (This limitation should be removed in a future version of the Mac OS.) In addition, MP tasks cannot contain or call any 680x0 code.

Because of this limitation, MP tasks must be compute-intensive—that is, the code in an MP task must use "pure" PowerPC processor instructions and not call either Toolbox or Mac OS routines. So far, Apple sees that only certain kinds of applications—those involving image processing, 3D rendering, digital video, or scientific and technical content—will run significantly faster if they are modified to be MP-savvy. Don't let these categories keep you from considering adopting Apple's multiprocessing technology—if your application has significant chunks of

compute-intensive code (or if you can modify it so it does), you should modify it to be MP-savvy.

API Overview

The Apple MP API contains the routines you need for creating, destroying, and interacting with MP tasks, including routines that

- determine whether the MP API is resident
- count how many processors are present
- create an MP task
- terminate an MP task
- let an MP task terminate itself
- let an MP task possibly yield execution to another MP task

In addition, the Apple MP API contains routines that enable you to manipulate

- message queues
- semaphores
- critical regions

You need these routines because of synchronization problems inherent in the Apple multiprocessing architecture. On the PowerPC processor, when a body of code writes to memory, the processor does not guarantee that the memory locations get changed in the exact order that the code specifies. You will use the message-queue, semaphore, and critical-region routines to ensure that when two MP tasks communicate with each other, the data sent by one MP task is actually available before the second MP task tries to use it.

Message queues are first-in, first-out (FIFO) queues. You can use them to tell an MP task what work to do and where to find the data needed to do it.

Semaphores store a value between 0 and some positive integer. You can use them to keep track of how many occurrences of a particular thing are available for use. You can also use them to let one MP task indicate when a resource is available to another MP task.

By marking a region of code as a critical region, you ensure that it cannot be executed by more than one thread of execution (either the application or an MP task) at a time. One use for critical regions is to ensure that non-reentrant code, which would execute improperly if two or more processes were to try to execute the code simultaneously, works correctly.

How Much Speed Can You Expect?

Obviously, the speed increase of an MP-savvy version of your application depends on many factors, including how much compute-intensive code it has and how much work you put into taking advantage of the Apple MP API.

However, I'll give you two data points to support the merits of the Apple MP API. According to one Apple source, experience with the Power Macintosh 9500/180MP (which has two 180-MHz PowerPC 604e processors in it) shows that compute-intensive code runs 1.5 to 2 times faster than it does on a single-processor computer running at the same speed.

The second data point comes from a review in the August 5, 1996 *MacWEEK*, titled "MP Options Deliver Low-Cost Speed Boost." The staff did benchmarks of five compute-intensive Photoshop functions. (Adobe Photoshop has plug-ins that use the Apple MP API.) The review reports that "multiprocessor Mac OS systems [with two 180-MHz PowerPC processors] cut one-half to two-thirds of the time for five compute-intensive Photoshop tasks." If you're interested in more details, the text (but not the benchmark) of this article is at the Web site http://www.macweek.com/mw_1030/rev_mp_opt.html.

Why You Should Adopt the Apple MP API

Although chip speeds have increased dramatically over the past several years, multiprocessing is the only truly scalable way to increase the amount of computational horsepower on the desktop. As MP options spread across the Power Macintosh and Mac OS-compatible computer product lines, every application—from entry-level consumer applications to top-of-the-line digital video solutions—will reap the speed benefit.

Assuming you have an application that includes compute-intensive code, there are two main reasons for creating an MP-savvy version of your application. First, there'll be a good market for your product, and second, you can make the transition with relatively little work.

The Apple MP API is important because it brings a single Apple-endorsed, hardware-independent API for multiprocessing to the Mac OS, which means that you can address the entire multiprocessor market with only one product. With the industry focusing on and supporting a single API, there will be increased interest in—and a larger market for—multiprocessing software.

Another factor that will ensure a good market for your MP-savvy application is the fact that

three major companies—Apple, DayStar, and UMAX—are or will soon be shipping Mac OS-compatible computers with two or more PowerPC processors. With multiple sources for multiprocessing computers, users will have multiple vendors from which to choose; this translates into more multiprocessing computers being sold and a better business proposition for you.

On the development side, the good news is that you can get into multiprocessing gradually. With only a modest effort (depending on how your code is structured), you can hand your compute-intensive code to the Apple multiprocessing architecture to get a modest speed increase in your application. Later, after you have gotten some experience with writing multiprocessing code, you can redesign your application (probably at a major X.0 revision) to get even more speed out of your application.

Several other factors will make the transition to an MP-savvy application easier. First, you can create MP-savvy applications with your existing compiler and build tools; in addition, the Metrowerks 9 compiler allows you to debug MP-savvy code (and other compilers are expected to follow with multiprocessing support). Second, you don't have to support two versions of your application—you can write it in such a way that it will work correctly with both single- and multiple-processor Mac OS-compatible computers.

What You Should Do

The Apple MP API has gone Golden Master, so you can begin using it without worrying about Apple making any incompatible changes to the API. You can also get the Apple MP SDK (software development kit), development information, and a list of current MP-savvy applications—as well as all the latest news on MP API support—at Apple's new Web site (<http://www.dev-world.apple.com/dev/mpdev.shtml>).

You can get the SDK from Daystar at <http://www.daystar.com/Developer/dev-pageone.html>. (On this page, it's labeled as the "Multiprocessing API"—click on the word "Download" to download the SDK.)

Apple will be offering multiprocessing "kitchens," in which attendees spend two days with Apple and DayStar engineers learning about the technologies and get one-on-one help incorporating the technology into existing applications. Apple just held the first kitchen in Cupertino, California, on September 19 and 20. For information on attending multiprocessing kitchens, contact Jason Wallace, Apple MP Evangelist, at mpevangelism@apple.com or 408-974-

5606.

Apple is currently gearing up its multiprocessing technical support, starting with the multiprocessing kitchens, and Apple is working closely with DayStar to provide technical support. You can reach Apple's technical support through the usual devsupport@apple.com address. You can also get technical support by writing to DayStar at mp@daystar.com.

In fact, DayStar recommends that you contact it *before* you start working on converting to an MP-savvy application. That way, its engineers can help you decide how best to modify your existing application using the Apple MP API.

The files supplied with the Apple MP API SDK enable you to create MP-savvy software on a single-processor computer, but you need to test your application on a multiprocessor computer before you ship it. DayStar has several options available to make this possible:

- developer discounts
- leasing and equipment loans
- checking your code

For all the above options, contact DayStar at mp@daystar.com. Apple is also offering hardware discounts to the members of its developer-support programs.

Future Plans for Multiprocessing

You don't have to worry about the MP-savvy application you create today being incompatible with future versions of the Mac OS. Apple has pledged that applications written to the Apple MP API will continue to work under future versions of the Mac OS (formerly known as *Mac OS 8*). In addition, existing MP-savvy applications will run even faster under the future version of the Mac OS that has Apple MP API support built into it.

On the marketing side, Apple is committed to making multiprocessing a mainstream feature of the Power Macintosh product line. This means that the installed base of multiprocessor Mac OS-compatible computers (from Apple and other vendors) will grow significantly in the future, presenting a larger market for your multiprocessing applications. Apple also plans some comarketing opportunities for the first show-stopping applications that use the Apple MP API well.

On the technical side, this same future version of the Mac OS will enlarge the scope of Mac OS multiprocessing by allowing MP tasks to include Toolbox and Mac OS routines. According to DayStar sources, this will possibly increase your application's responsiveness but not its

More Advantages of the Apple MP API

From among the different ways that multiprocessing could have been implemented on the Mac OS, Apple chose the Apple MP API. Here are some of the reasons (in addition to the ones mentioned in the body of the article) that the Apple MP API is faster and easier to use than competing designs:

- The Apple MP API specification stipulates that the processors in an MP-compliant computer must be cache-coherent—that is, you don't have to worry about the possibility that data in one processor's cache doesn't get written to main memory before another processor attempts to read the memory locations affected. This makes writing multiprocessing applications much easier than it would be for the simpler multiprocessing architectures that don't guarantee cache coherency.
- All of the attached processors in a multiprocessor Mac OS-compatible computer are on the processor bus, not a slower auxiliary bus (such as the NuBus™ or PCI bus). This makes the Apple MP API superior to earlier accelerator-card solutions, which were slowed down by the bandwidth limitations of the bus system used.
- All applications and MP tasks have access to all the memory in the system. This makes the Apple multiprocessing technology faster and easier to program than those that require the overhead of having to copy memory areas to a task-controlled memory area before they can be accessed.
- In the Apple multiprocessing architecture, all MP tasks are preemptively multitasked; in addition, if an MP task is waiting for some resource before it can execute, the Apple multiprocessing architecture does not execute it until the required resource becomes available. This prevents the attached processors from wasting time on an MP task that is blocked from doing useful work.

speed. (Toolbox and Mac OS routines are not particularly compute-intensive.) The increased responsiveness will come from the MP computer's ability to execute tasks—for example, menu drawing, window drawing, and I/O—on separate processors while the main processor continues to do useful work. The increased speed will still

come from compute-intensive code handled by the Apple MP API.

Multiprocessing Is the Future of the Mac OS

People are buying multiprocessor Mac OS-compatible computers today, and as soon as they do,

they start buying applications that make use of multiprocessing. As time goes on, the installed base of multiprocessor Mac OS-compatible computers will increase. For these reasons, you should consider making your compute-intensive application MP-savvy—it's not difficult, and applications that support multiprocessing will have an advantage over competing applications that don't. ♣

FEATURE

Fear and Coding in Dearborn

By Dave Johnson

Never Attended MacHack? Here's the Next Best Thing

Thursday, June 20, 1996, 12:28 A.M. Half past Wednesday. I've just arrived, and I'm standing in room 339 of a Holiday Inn, in a nondescript section of a nondescript suburb of a large Midwestern city in Michigan. Looking out the window I could be anywhere in the United States: There are a couple of fast-food restaurants, a freeway, a strip mall. The lack of culture, or rather the plain vanilla flavor of what culture there is, is palpable. But it really doesn't matter much. The conference I'm here to attend

brought its own culture along with it. This is MacHack 1996.

MacHack is a Macintosh programmer's conference, and it attracts some of the best. It's much smaller and more focused than Apple's huge, flashy, expensive Worldwide Developers Conference and much less evangelical.

The sessions and talks range all over the map, from how-to's covering specific technologies (both new and old), to discussions of the pros and cons of different programming languages, to workshops on how to survive as a contract programmer. The crowd is enthusiastic and smart, and fiercely committed to the Macintosh. It's a lovely group to be around, especially in these troubled times, when even people who should

know better are wondering if the Mac is dead. Ha!

The conference was scheduled to begin at 12:01 A.M., just after midnight, an indicator of the programmer's culture that pervades the entire event. Many events during the conference are scheduled for late at night, and very few in the mornings. Caffeinated beverages of every sort are freely available at all times. The machine room, a big conference room full of Macintosh computers for people to use, is open around the clock. Every hotel room has a data port. The hotel maids have been warned not to knock on any doors before 10:00 A.M., and the conference attendees have been warned that if we want our rooms cleaned, we have to get up by 2:00 P.M.

I bet this Holiday Inn hasn't seen a convention like this one.

Flashback to the start of my trip: Wednesday, 1:32 P.M., California time. I'm in a shuttle on the way to the airport, my bags stashed in the back. A question in my mind is whether I'll write a hack for the contest. A centerpiece of the MacHack conference is the hack contest, wherein contestants write software that often modifies the system in hilarious ways, but I'm not at all sure I'll participate. I'm ready, just in case: I brought a variety of crusty old code with me—some juggling software, an evolution simulation I wrote years ago, a flocking algorithm, some fractals—all my favorite recreational code.

But I don't know—I'm older and less adaptable now than I once was, and somehow the prospect of staying up all night writing amusing but useless code—something that used to fill me with gleeful anticipation—now fills me with a sick dread instead. I want my nine or ten hours of sleep every night. Without it, I'll just get cranky. Then again, the old creative urge is still strong, and of course I'd love the glory of it all, to get some applause from my colleagues, my peers. I still want to do it, I'm just not sure I can face the strain; I know all too well the toll it will take. We'll see.

Thursday, 1:14 A.M. We're about a half-hour into the keynote speech, which started late. I'm not too impressed with the speaker. He's mostly spouting his opinions, which I don't find particularly interesting. Looking around at the audience members, about a third of them are doing something else: chatting quietly, typing on PowerBooks, going through the conference materials. I guess I'm not the only one who's a little restless.

Thursday, 3:32 A.M. I'm back in my room. I can't remember the last time I was awake at this hour. After the keynote let out, I milled around a bit and talked, then I came up here thinking I'd get some sleep. The only problem is, I'm totally wired. I suspect it has something to do with the soda I drank before the keynote, something called *Mello Yello*, apparently a Midwest thing. (I have a weakness for odd soda pop—"odd" meaning I haven't seen it before or it's uncommon, like Cactus Cooler or Kickapoo Joy Juice.) I glanced at the ingredients on the Mello Yello can, and caffeine was listed, but damn, there must have been a lot of it. Later, I find out that it's basically Mountain Dew in a different package. Well, there's no way I'm sleeping anytime soon, so I think I'll head back down, maybe even work on some code. What a scary thought.

Thursday, 9:33 A.M. The wake-up call was *painful*. I finally got to sleep around 4:30 or 5:00. I wanted to make a 10:00 A.M. session, so I arranged for a 9:30 wake-up. Lordy, I'm too old for this. Thankfully, the Holiday Inn has thoughtfully provided an in-room coffee maker, so I drink some lousy coffee. The shower is good and hot, and it even has a Water Pik Shower Massage that pummels me back to reluctant consciousness and humanity.

Thursday, 12:18 P.M. It's amazing what a restorative effect caffeine and food can have—at least for a while. The hotel coffee isn't half bad. I looked in on a couple of sessions this morning ("Be OS Overview," "Starting With PowerPlant"), and have spent some time just wandering around, talking to folks, collecting opinions about *develop* magazine.

This Holiday Inn has a big central atrium, the ceiling several stories above, natural light pouring in during the day. It's littered with round tables with chairs, high-quality fake trees that are kept well dusted, and soft chairs and couches for lounging. There's also a sort of fountain and pond dividing it in half, providing a nice white-noise background sound. It's the default center of activity for the conference (along with the machine room, which is just off the atrium). People can always be found there talking or sitting with their PowerBooks. All the sleeping rooms are well removed from the center, making it OK to make noise there at any hour. Networking cables snake their way out of the machine room and across the carpet. There's even a network router sitting on the wood chips under one of the plastic trees, and the table closest to it is very popular.

I also spent a little time this morning working on my flocking program. This is the application version of a screen saver I wrote several years ago that simulates the flocking behavior of birds. It's based on an algorithm presented at a computer graphics conference, and I'm hoping to use it for my hack—that is, if I actually decide to write one. I'm still hesitating, even as I tune up the code a bit in preparation.

Thursday, 8:43 P.M. "Bash Apple" is in full swing. This is the session in which every Apple employee at the conference gets up on stage and everyone else gets a chance to bring up their issues with Apple Computer. The issues range, as you might expect, from personal whinings to legitimate complaints to insightful suggestions. As you might expect from a crowd of this caliber, they are heavily weighted toward the latter. The issues will later be voted on by all attendees,

concluding in a top-ten issues list to be taken back to Apple for action.

I missed most of the afternoon sessions, talking to folks and working on my code. I had a session of my own scheduled, "Juggling for Programmers," at 3:00, but no one seemed to know about it. I put up some signs, so hopefully the turnout tomorrow will be a little better.

Thursday, midnight or thereabouts. Pizza in the atrium again, just like last night before the keynote—stacks and stacks of flat cardboard boxes filled with greasy deliciousness, something like 50 extra-large pizzas scattered about. In fact, there's free pizza in the atrium every night at midnight, making lactose tolerance a prerequisite for this conference. I'm tired. I'm going to bed. Well, maybe just one more piece.

Friday, 9:07 A.M. Fifteen hours 'til the hack contest. I've done a little coding, but I'm still not sure I'll enter. However, I got plenty of sleep last night: I passed out around 1:30 A.M., slept right through to my 7:30 wake-up call and another at 9:00, so I'm feeling fresh and alert. At this hour, the machine room is practically deserted, so I pick a nice machine, a Power Macintosh 8500/120 with a 17-inch screen, and I settle in to see if what I want to do is really possible.

There are several sessions I want to go to today: "Mac OS 8 Graphics and Printing," "Debugging Environments," "Pippin Under the Hood," "Bash Metrowerks," and a couple of others. It's going to be a busy day.

Friday, 12:23 P.M. Less than 12 hours to the contest. I haven't moved from my machine. *I think I can pull off my hack.* The idea is to get the icons in the front Finder window to detach themselves from their names and go flocking around the screen. With the help of Apple's new Game Sprockets (DrawSprocket, to be exact) and the Scriptable Finder, I think I can do it. I've spent much of the morning learning more than I ever wanted to know about Apple events, so that I can query the Finder to get the icons and their positions. Greg Anderson's sample code (from his article in *develop* Issue 20 about the Scriptable Finder) is proving invaluable.

The machine room is really hopping now, people leaning into their machines looking determined, consulting with each other, typing and mousing feverishly. I've blown off the morning sessions I was planning to attend, and I have a feeling I'll spend most of the afternoon here too. That old obsessive feeling is back, and I'm spurred on by all the happy coders around me.

Social programming is a really great experi-

ence, far preferable to doing it alone. I've only experienced it once before, at an OpenDoc kitchen, and I'd forgotten how fun and infectious it can be. One very nice thing is that when you run into a problem there's a SmartFriend within earshot who's willing to help, so progress is much faster. But more than that, the feeling in the room is electric—all those minds bent to their tasks, all that humor and creativity unleashed. If you've never had the experience, you should.

Friday, 6:17 P.M. I stand up and stretch. My shoulders are tight, my eyes feel crusty and irradiated, but my mind is going full blast. Progress has been slow but steady. Those damn Apple events were giving me fits, mostly due to my ignorance but partly due to the strange types and data structures they use. I think I have it under control now. In Apple-event land, type coercion is definitely your friend.

Sure enough, I blew off all the conference sessions. I can't stop coding. Except for juggling, of course; since it was my session, I thought I'd better attend. Lots of people came out of the woodwork—four or five beginners learned for the first time, and lots of others learned something new. All in all, a success. (OK, I admit, bringing 85 juggling balls was a bit over-optimistic, and I'll keep it to 30 or so the next time.) Juggling was a welcome break from coding, too; it's nice to stand and move around a bit after sitting glued to the screen for five hours. But after a couple hours of juggling and teaching, I was ready to get back to it. I'm hooked on the hack, and it's going to be a race to finish on time.

Saturday, 12:42 A.M. The Hack Show has begun, and I'm still trying to clean up my code and get it in. It's working well, but there's a lot of embarrassing garbage left in the source code, vestiges of the old code I adapted it from. Since the code will be distributed publicly, I want to get it cleaned up as well as I can. Laughter and applause erupts periodically from the big room, and I'm typing as fast as I can.

Saturday, 10:00 A.M. I manage to pry my gluey eyes open, and the light coming into the room is painful. I didn't get to sleep until almost 6:00 A.M.

When I got up on stage to demo my hack the first time around 1:00 A.M., it wouldn't launch: The machine didn't have some software installed that I needed. I finally got to show it to an exhausted and thinned-out crowd just before 5:00 A.M., right near the bitter end, and thankfully

it worked just fine. The crowd applauded wildly—well, as wildly as can be expected at that hour—and it was very gratifying.

The Hack Show was very long, but lots of fun. The hacks were all over the map; some silly, some deep; some hilarious, some not; some brilliant, some trivial. It was really fun to see the kinds of things people came up with. One hack replaced all the decimal numbers in the system with Roman numerals. At "Bash Apple," an issue that came up was the ridiculous numbering scheme for recent system releases. System 7.5.3 Revision 2? Please. So a couple of Apple folks entered a hack that made the system version numbers in the "About this Macintosh" box into editable text items, so you could enter whatever numbers you liked. You get the idea. Tons o' fun.

Saturday, 5:02 P.M. Once again caffeine is my savior. I went to a couple of sessions today and also spent a lot of time back in the machine room, improving my hack for posterity. I wasn't the only one, either. It's funny how once you get going on these things, it's hard to stop until they're done. And as we all know, software is never really done, just shipped. So even when it's just for fun, it's hard to know when to stop.

In a couple of hours is the awards banquet, when they announce the results of the "top issues for Apple" voting and the winning hacks. Everybody has a chance to vote on the top issues and the hacks; it's very democratic. It'll be interesting to see what bubbles up to the top.

Saturday, 9:22 P.M. The banquet is over, and everyone's getting ready to go to a movie (another in a long series of MacHack traditions). Tonight we're seeing *Eraser*, the latest Schwarzenegger adventure film. Afterwards will be an ice cream social.

I was very pleased at what was voted the number one issue for the top ten: encourage new Macintosh programmers. There are a zillion young, enthusiastic, fledgling programmers in high schools and colleges all over the world, and Apple would do well to help them out. Almost all the Macintosh programmers I know started in high school or college, usually on their own, teaching themselves from an old copy of *Inside Macintosh*. If Apple had been there to help, how many more would there have been?

The winning hack was the one everyone knew would win: called *Booting Gallery*, it replaces the old, boring march of startup icons across the screen with a game! Actually, two possible games, selectable with the Caps Lock key. In both cases, the icons of the extensions

become the targets in an arcade shoot-'em-up. Startup will never be the same.

I'm also happy to report that my flocking hack was voted third place. I won a bird feeder.

Sunday, 1:31 P.M. I'm home. I stayed up practically all night last night improving my hack, while around me in the machine room, everyone played Marathon or Bolo into the wee hours. And I'm about to start up my machine here at home to work on it a bit more.

I'm exhausted. Totally fried. I need to sleep for a week. My head is buzzing. My brain hurts. My eyes hurt. My fingers even hurt, from typing too much. I never want to see pizza or potato chips or caffeinated soft drinks again.

I'd do it again in a heartbeat.

I'm already planning to go next year. I probably should have been going to this conference for years now, but somehow never got to it until this one, the eleventh annual. I doubt I'll ever miss another as long as I'm still programming the Macintosh.

It was amazing: the camaraderie, the shared passion for the Macintosh, the friendly competition for best hack, the technical excellence of the papers and sessions. And the incredible collection of incredible minds—that perhaps more than anything else. Hanging out with a crowd of that caliber can't help but be inspiring.

If you're a Macintosh programmer and you've never been, go. If you have, then you already know exactly what I mean. ♣

Dave Johnson is the technical editor for Apple's quarterly programming magazine, develop. (Actually, he's listed as the magazine's "technical buckstopper" on the develop masthead.) He's interested in juggling, dogs, and arcane things. If you'd like to try out Dave's back (it's really worth it), start at the Web page <http://www.machack.com/Resources/index.html>. Go to the MacHack 96 Hack page and download FINDERFLOCKS.

CD HIGHLIGHTS

System Software/SDK Edition

continued from page 15

into a SOM-based shared library to reduce the size of resulting sample parts and provide better code reuse. Currently, this library contains most of the Foundation and OS layers. As development continues, Apple will attempt to move more of the framework into this shared library. Because of this move, this release requires that your parts be dynamically linked to the ODFLibrary.

This release supports the following development environments for PowerPC:

- Metrowerks CodeWarrior 9
- MrC for MPW
- Symantec C++ for Power Macintosh 8.0 (Rainbow)

(Note: Although projects for the Symantec C++ IDE are provided, they will not build correctly. Apple is working with Symantec to resolve those problems and will post an update on the Web as soon as possible.)

This release supports the following build environments for 680x0:

- Symantec C++ for MPW
- CodeWarrior 9 for 680x0

For specific instructions on configuring your compiler for use with ODF, see the Getting Started folder.

PCI Driver Development Kit 2.0

This folder contains information on designing PCI cards and drivers for the PCI-based Power Macintosh computers, such as the Power Macintosh 7200, 7500, 8500, and 9500 computers. This version of the driver development kit contains updated samples, newer headers and libraries, a new tokenizer, an update document to augment *Designing PCI Cards and Drivers*, and an updated version of the PCI Binding document (version 1.5).

Snippets Update

This folder contains new snippets from DTS:

- *CreateDirWithCustomIcon*. This snippet

demonstrates how to create a directory and steal a custom icon from another directory by using the CatMove routine to move the Icon* file from the other directory.

• *FileSharingOn*. This snippet shows how to determine whether file sharing is on.

• *FormatAsDOS*. This snippet demonstrates how to use the newer Disk Initialization Package routines DIXFormat and DIXZero to format a floppy disk as a DOS disk. DIXFormat, DIXZero, and DIRReformat are documented in the "Inside Macintosh: Files Errata" technical note.

Please note that this snippet does not provide any prompts. It merely takes the floppy disk in the first floppy disk drive and reformats it as a DOS disk with the name *MS-DOS Disk*. The purpose of this snippet is to demonstrate the new API.

• *IsPCExchangeInstalled*. This snippet shows how to check for the existence of Macintosh PC Exchange. The FSMGlueLib.o file is available on the Mac OS SDK CD in the File System Manager Libraries folder.

The previous four snippets run under Metrowerks CodeWarrior 8 with the Universal Interfaces 2.1 from E.T.O. #18.

• *ictbSample*. This snippet shows an example of the item color table ('ictb') resource, which is discussed in *Inside Macintosh: Macintosh Toolbox Essentials*, page 6-158 to 6-164. It is based on the "Dialog Manager Q&A" technical note.

The item color table resource is difficult to manipulate by hand. This sample provides careful documentation of the process; it shows how to use different colors and fonts in EditText items, and how to use color in buttons.

• *TimeZone.Daylight*. This program demonstrates how to use the ReadLocation function to show the current location and time-zone information that is in the Date & Time control panel. The program also shows how to determine whether Daylight Saving Time is turned on or off.

SOMobjects for Mac OS

Apple has licensed and ported IBM's SOMobjects™ technology to the Macintosh. SOMobjects is a multiplatform standard providing for system-level sharable objects in a language-neutral way. It also

solves the "fragile base class" problem of object-oriented shared libraries, making it unnecessary to recompile client libraries when the base class they inherit from is in a different library and is changed.

To offer these benefits, SOMobjects requires the use of an Interface Description Language (IDL) compiler as a first step in the development process. An MPW-hosted version of this compiler is now available.

SOMobjects runs on top of the Code Fragment Manager (CFM), and thus is available for both 680x0 and native PowerPC processor-based Macintosh development. It is a foundation technology for OpenDoc.

SOMobjects for Mac OS version 2.0.8 contains several bug fixes over version 2.0.7.

WorldScript Updater

The WorldScript Updater 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 WorldScript-dependent applications should install these pieces, since they improve system stability and performance; QuickDraw GX and Power Macintosh users should upgrade to these pieces, because they provide important improvements over previous versions. The files in this WorldScript updater are taken from the Arabic Language Kit 1.0.1, Hebrew Language Kit 1.0.1, and Cyrillic Language Kit 1.0; if you are running one of these kits and do not also have Japanese or Chinese installed, you do not need the WorldScript Updater. For details, see the document WorldScript Updater Read Me.

Alex Doshier
Developer CD Leader and
Online Content Librarian

Business

Packaging Feature: Your retail software box design must stand out on crowded shelves, and your box copy must sell your product without the help of sales people.

Comarketing Feature: Apple is expanding and organizing developer comarketing programs in an effort to help you more effectively market your products.

Creating Packaging That Sells

By Valorie Cook Carpenter,
Market Savvy Consulting Group

If you sell your product through retail distribution channels, the appearance of that product on the shelf is critical to its sales success. A great package helps you sell your product to retailers and customers alike; conversely, a poor package can keep your product from reaching its full sales potential. To help you create a package that works its hardest to sell the product inside, here are nine tried-and-true steps that I've put to good use in companies such as Brøderbund and Procter and Gamble. These steps are based on my marketing experience of over 20 years in both software and the consumer packaged goods industry.

Step 1: Select a Name That Helps You Sell

Selecting a poor product name probably won't prematurely kill a product, but a good name definitely helps sales. So why not attempt to create a compelling name for your product?

First things first. While your product is in development, give it a nondescriptive code name—one that couldn't possibly become the final name for the product. This is important for a number of reasons.

In the early stages of development, a meaningless name lessens the possibility that a competitor will understand the essence of your product concept if it is overheard while your staff is discussing it at lunch, on an airplane, in the lobby of a retailer, or at a trade show. It also keeps your team from "locking in" a name too early, before the marketing group really understands the key differentiators of the product from the target customer's point of view. These differentiators should guide the naming and positioning of the product

to maximize its effectiveness. Development groups get very attached to names, even nonsensical ones; don't compromise your marketing effectiveness by giving your product anything but the name that best communicates your key message to your target customers.

Also, make sure that your product's code name won't embarrass the company or get it into legal trouble if it appears in print. Names based on neutral themes (animals, rock types, and so on) work well. Strive for whimsy and charm rather than grotesque or just plain gross names.

As a prelude to selecting a final product name, develop a strategic objective statement. This is a short, half-page document that summarizes a product definition (in one or two sentences), a target customer, and a "why-to-buy" customer message (ideally expressed in four to seven words).

In this document, set high-level priorities for the new product name. Typically, great product names are

- short
- related to the product's functional purpose
- easy to pronounce and understand
- easy to spell
- memorable
- localizable

Some excellent Macintosh product names that I've run across include Datawatch's Virex (for preventing and exterminating computer viruses), Iomega's Zip drives (fast, convenient data storage), and Intuit's QuickBooks (quick business accounting). Also, I admire Berkeley Systems' tongue-in-cheek CD game show title, *You Don't Know Jack*, for its humor and audience appeal (though it's not particularly short).

Some why-to-buy statements that I find particularly compelling include "Organizes Finances Painlessly" for Intuit's Quicken,

and "The Paint Program Just for Kids" for Brøderbund's Kid Pix.

Your strategic objective statement should be finalized and approved by all interested parties at least *eight months* before the product is scheduled to release. This lengthy time frame allows for the inevitable dead ends and roadblocks that occur in the naming and package design process, without burning people out.

There are several approaches you can take to develop an effective product name, once you have a strategic objective statement that everyone agrees upon. The two I'll discuss here are brainstorming and hiring an outside agency.

In-House Brainstorming

Invite a small group of 10 to 15 creative people to a one-hour brainstorming session. It's preferable to schedule it in the morning, when people are fresher. If there are lots of interested people, hold several smaller brainstorming sessions rather than one huge one.

If you're organizing the session, here are a few tips for achieving the best possible set of potential names:

- Give the participants a copy of the approved strategic objective statement in advance, and ask them to think about possible names before the meeting.
- Bring dictionaries and thesauruses to the meeting.
- Appoint a leader for the brainstorming session, and have that person set the ground rules in the beginning. Emphasize that the objective is to generate *lots* of names, and that none of the names will be evaluated or judged until later. Write down all names (even names like *RoadKill* and *Spam-Master* are fair game!) on a large pad of paper.
- Tape all the filled sheets of names on the walls of the room. Try to come up with

at least 100 names.

- At the end of the session, strive to get consensus on five to ten of the best names.
- Type up the entire list of names and high-light the favorites.
- Repeat as necessary until a handful of preferred names emerges.
- Obtain trademark clearance on the top five names. (By the way, a quick name search on the Alta Vista or Yahoo Internet search engines can save you time and legal fees.) This is probably the most frustrating step, as many great names are already taken.

Hiring an Outside Naming Agency

Hiring a naming agency is expensive (it can cost \$25,000 or more) and often these firms only commit to providing you with one trademarkable name. Large companies with the resources to go this route, however, have obtained outstanding names in this manner. It may be worth obtaining outside bids for these services, particularly if you're unsatisfied with the results of internal brainstorming.

Step 2: Design for Key Channel Partners

The number of outlets carrying personal computer software products has grown to more than 30,000 storefronts over the past few years in the United States alone. This explosive growth has come primarily from new, nontraditional chains entering the marketplace, such as Wal-Mart and Best Buy.

Computer superstores (such as CompUSA, Computer City, and MicroCenter), office supply superstores (such as Office Depot and Staples), consumer electronics stores (such as Best Buy and Lechmere), warehouse clubs (such as Price Costco), mass merchants (such as Target and Sears), and others have all joined computer specialty stores (such as Egghead and Electronics Boutique) as sources of computer hardware, peripherals, software, and related supplies. Each of these account categories attracts a different type of customer and needs to be analyzed specifically for its packaging needs. However, in general, they all share one characteristic: They require more of a self-service, "grocery-store mentality" in packaging design.

Impact of Distribution Trends

The increase of self-service retail stores has significantly affected the approach that software companies need to take in marketing, merchandising, and manufacturing products.

Here are some distribution trends and resulting implications for your manufacturing teams:

- More inventory is required due to the greater number of stocking locations, so it will save you warehousing and shipping costs if you can minimize packaging weight and volume without compromising the package's competitiveness on the retail shelf.
- As the number of copies of your product in the channel increases, you should expect more returns. You'll want to make sure that your software packages and shipping containers are strong enough to survive additional shipping, and that the increased cost is taken into account in your budget.

• You must ship directly to stores—at multiple locations—rather than to just a few centralized warehouses. This may mean you'll need to include fewer units per shipping container than you did in the past, or even offer multiple case packs, as the packaged goods industry does. Again, you will have to take into consideration the incremental costs associated with this reality.

- Pilferage, shrinkage, or slippage (in other words, theft) is much more of an issue; packaging needs to be specifically designed to foil thieves, or at least slow them down. Many software publishers distributing their programs on CD-ROM have created cardboard inserts that hold the jewel case more securely in the outer package for this reason.

The following are some of the trends affecting the marketing and merchandising of software:

- The retail store's staff is less knowledgeable, so there is more reliance on packaging and merchandising to sell the product. You'll need to compensate for this with better box copy, graphics, and point-of-purchase materials.
- The self-service environment increases the need for high-impact packaging and merchandising, especially amid the clutter of competing products.

Step 3: Evaluate the Competition

It is vitally important to design your packaging in the context of the products that it will be competing with on the retail shelf. The most striking design or color will not draw attention to your product if the product right next to it looks too similar. That's why I strongly recommend that you visit the retail storefronts of your key channel partners to evaluate competitive products that are already on the shelf. What's more, you should bring home the closest competitors' packages to use as the "benchmarks to beat" in the design process.

Also, don't make the common mistake of assuming that your product has no competition. Even if it doesn't have a head-to-head competitor, it will sit on the shelf next to other related products. It's in your best interest to *anticipate* the product section that it will be placed in, then design your package to stand out. This is much better than leaving this critical decision to your channel partners.

Step 4: Set Strategic Packaging Objectives

As with naming the product, the best starting point for creating an effective package design is to develop a strategic packaging objective statement. Again, it should be short (no more than one page) and should cover the following key points:

- the product name
- the target customer
- the "why-to-buy" target customer message—again, ideally expressed in four to seven words
- a *concise* description of the product (one or two sentences at most)
- any information that absolutely must go on the package, such as a platform designation, system requirements, and so on
- the tone, or the emotions the package must evoke in the target customer; for example, corporate and business-like, warm and inviting, or whimsical and fun
- any other information that would be helpful to the creative team

Step 5: Select Creative and Collaborative Designers

Should you use an-house or outside design firm? Either can work, but what's important is that the creative team can work both creatively and collaboratively to meet strategic objectives.

Here's one way to manage the process of selecting an outside agency:

- *Solicit recommendations.* Identify packages that you think are effective in the retail environment, then find out who did the graphic design. Sometimes the designer is even listed on the package (in small print on the back or bottom panel). Otherwise, if the software publisher isn't a direct competitor, they may be willing to tell you who did the design. Initially, identify three to six prospective design firms.

- *Meet with each potential design firm.* Ask each design firm to present their portfolios. See if they mention design objectives of past clients and ask them if they have data that proves that their designs resulted in increased sales. Discuss how they like to work, and make sure they're willing to work to a strategic objective statement, within a competitive framework. Ask them how they've resolved creative differences with clients in the past, and make sure they'll present you with a number of design alternatives. Be wary of agencies with a particular "look" unless you're sure you want that look for your product. Having designers with some software package design experience is ideal, but not absolutely necessary.

- *Select the most creative and collaborative design firm.* If forced to choose between creative and collaborative skills, go with the latter. You'll be more satisfied with the results!

Although most graphic design firms work project by project, rather than on a ongoing retainer, it's preferable that you work with a single firm for all of your package design work. There's a learning curve for every new agency that comes on board. Once you've found a group that understands your products and company, repeatedly using this group will result in a smoother design process and better results.

Step 6: Aim for Maximum Shelf Impact

The whole point of the front of your package is to get it noticed: You want to attract target customers as they're walking down an aisle full of software, and you want to motivate them to pick up your package. *You must accomplish this in less than two seconds, from as far away as ten feet.* That

requires nothing less than maximum shelf impact!

Here are some design elements that contribute to high shelf impact:

- *Bold, clear colors.* Avoid black, because it turns into a "black hole" on the shelf, especially when shrink-wrapped. Two well-known software products that have suc-

You must attract target customers in less than two seconds, from as far away as ten feet.

cessfully "owned" the bold colors used on their packages are Lotus 1-2-3's goldenrod and Kid Pix's bright green.

- *Simple graphics that relate to the product.* Don't use "insider" visual jokes, photographs of founders, and so on. Norton Utilities, which once featured a picture of Peter Norton on the box front, is an exception to this rule because of Peter's fame among computer users.

- *Large, readable fonts.* Avoid "reverse-out" type (in other words, white or light-colored type on a dark background), because it's harder to read than dark type on a light background. The fact that Claris, Microsoft, and Apple (companies that, by the way, spend a lot on customer research) stay with the black-copy-on-a-white-background approach, is a testimony to the effectiveness of this rule.

- *Limited copy.* You only have a few seconds to sell your product, so make sure your box copy is short and compelling. A good example of just the right amount of copy can be found on the products from Living Books (for example, *Just Grandma and Me*).

A word about flaps—those extra pieces

of cardboard that open out like a greeting card on the front panel of some software packages. Don't use them—they can compromise your shelf impact by yawning open and obscuring the front panel you've worked so hard on. Further, they usually allow the product copy to run on far too long. If you must use a flap, put it on the back panel, where it won't affect your shelf impact (because it can't yawn open), and make sure that it is inviting to look at and read—use lots of graphics and a minimal amount of copy.

Step 7: Manage the Design Process

Once you've finalized a trademarkable name and gotten the strategic objective statement approved, design can begin.

Before you conduct the first creative team meeting, you should select a packaging structure. Choose a packaging structure based on your product's competition as well as the shelving and shipping requirements of your key channel partners. Their input may influence whether you choose a folding carton, candy box, or slipcase box format, or whether you need to limit your package dimensions, material, or weight. In addition, make sure that your proposed structure is sturdy enough to withstand the rigors of worldwide shipping, and that it communicates appropriate value to your target customer when it's picked up.

I learned a lot about perceived value when I worked for Ashlar, a company that sells an award-winning \$995 CAD software product called *Vellum*. For this type of product, a \$1 plastic CD jewel case wouldn't be appropriate. To look like a \$995 product, the box needed to have some physical heft (a good use for reference manuals and tutorials), as well as larger dimensions.

Choose a box size that will fit on your key channel partners' shelves and is comparable to other products of similar functionality and price. And finally, by working with packaging materials vendors, make sure to build and shrink-wrap a prototype package that is packed with the actual media and manuals (or a manual mock-up that includes the same page count, binding, and paper type). Test it for balance on the retail shelf, making sure it won't tip over.

First Creative Meeting

Once you've decided on the physical configuration of the box, it's time to bring in the graphics creative team. Insofar as is possible, have all interested people attend this initial meeting. Here, the marketing group should accomplish these tasks:

- Present the strategic objective statement, bringing new creative team members up to speed.
- Show competitive product packages, both in the meeting and ideally in an actual store setting, such as a computer superstore. This is the environment in which your product must stand out to achieve shelf impact.
- Review the proposed packaging structure.

Second Creative Meeting

Again, try to have all interested parties attend this meeting, at which the design group presents a number of rough design directions (at least three, and preferably six to eight) for your box's front panel. Competitive product packages should again be presented for shelf context. Participants should then narrow the choices for further development to the best two to three *based on the strategic objective statement* and their potential for high shelf impact in the retail environment. Or, ask the design

group for more rough concepts. Strategic feedback, rather than art direction, is the key to successful refinements of the designs. For instance, offer your designers advice such as "The product name needs to be more prominent" rather than "Make the product name red."

Third Creative Meeting

Here the design group presents the refined concepts, again in the context of competitive packages and the strategic objective statement. These designs should also be viewed in a retail store environment. Hopefully a clear winner—in other words, the most visible package on the retail shelf that meets the requirements laid out in the strategic objective statement—emerges for final development.

At this point, all panels should be detailed. Make sure the product name and why-to-buy message appear on every panel—the front, back, both spines, top, and bottom. If you're using folding cartons, order a "tuckable" bottom-flap configuration rather than a center-cut bottom, because this will give you a larger printable surface area.

Step 8: View Proposed Designs in Stores

As you may recall, in the beginning of this article I suggested visiting your key channel partners' storefronts to check out the competition. Once you have a tight prototype of your final design, or better yet, two or three concepts, take them—and your design team—into one or more stores. Put the prototypes on the shelf in the section you have targeted for their placement, stand ten feet away, stroll down the aisle, and see how they actually look next to their competition. It's always an eye-opener—trust me! If your designs don't physically fit on the shelf or fail to leap out at you, it's back to the drawing board for another round of concepts.

Step 9: Allocate the Time and Money to Do It Right

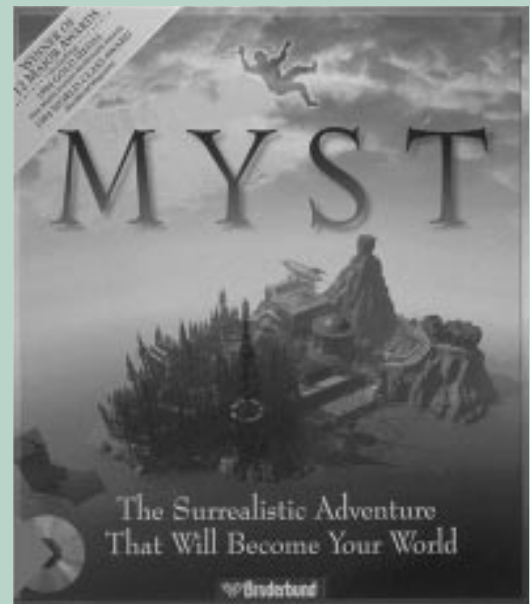
I like to start package design at least six months prior to a product's scheduled release date. Again, it can be done much faster, but for the most part, compressing the package design cycle is unnecessary; it adds stress to an already stressful situation, and it results in compromises. In terms of cost, I would suggest budgeting \$25,000 to \$50,000 for the complete process, from design through mechanical art preparation (the last step prior to actually printing the boxes).

The Making of the *Myst* Box

When my marketing group at Brøderbund began working on the *Myst* package, we hired an outside design agency. Based on our strategic objective statement, the agency generated a number of striking front panel concepts—none of which the senior marketing manager for the product felt accurately represented the product. After numerous, increasingly frustrating rounds of creative work, the program's outside developers finally submitted a front panel design based on the central computer graphic of the game—Myst Island. With some minor modifications, this became the final design, despite the fact that it was a very recessive package on the shelf—it used murky, grayed-out colors, a computer graphic rather than a more traditional illustration, the product title in a less-than-optimal position, and the why-to-buy statement in reverse-out type. In fact, we had created a similar design with much more shelf impact, due to a pink-orange sunrise in the background behind *Myst* Island. But we didn't go with it because the marketing staff convinced me that the package they preferred was truer to the game. Over 2,000,000 copies later, I have to say they were right!

So what did I learn from the process?

- The steps I've detailed here are guidelines that need to be applied intelligently in each specific situation. Sometimes other considerations outweigh the goal of maximizing shelf impact.
- If you're going to "break the rules," do so deliberately, carefully weighing what you are giving up and risking versus what you hope to achieve.



Apple Expands and Organizes Developer Comarketing Programs

By Kris Newby, Apple Directions staff

One of the advantages to having ex-developers, such as Heidi Roizen and Guy Kawasaki, in influential positions at Apple Computer, Inc., is that you know there's almost always someone within earshot of top management to champion developer issues. With the important holiday selling season just around the corner, Apple Developer Relations—with the help of these developer advocates—has been working to create more and better ways to let Apple computer users know about your products. In this article, we present a comprehensive list of Apple-sponsored comarketing opportunities that you can take advantage of, most of which are rolling out just in time to help you plan your holiday product promotions.

On a long-term basis, Apple Developer Relations (ADR) wants to make sure that you

know where to go within Apple to take advantage of the comarketing opportunities that it makes available. Since Apple is such a large, distributed company, the Third Party Marketing group in Apple Developer Relations is taking on the job of combing through the numerous marketing, product marketing, and sales organizations to provide vital comarketing information in two places (see the “Key Contacts for Comarketing Opportunities” box):

- the ADR Comarketing Web page, for ongoing opportunities
- Apple Developer News, for late-breaking and one-time opportunities

While you may still have to work with Apple through a number of Apple contacts and organizations, these two sources will make sure that you always know what opportunities are available to you.

Organizational Changes Help Developers

From a marketing perspective, three recent Apple organizational changes bode well for developers. First, with Apple moving from CPU-focused marketing groups to customer-focused groups, there's more of an emphasis on providing our mutual customers with the best integrated hardware and software solutions for target markets. As a result, we're seeing a wider range of third-party products being promoted and bundled with CPUs. You can see this “total solution” approach at work in Apple's Internet group, with products such as the Apple Internet Server Solution bundle. Or you can see it in the small office, home office (SOHO) group, with their sponsorship of the *MacHome Journal Digest CD Sampler*, which will ship with *MacHome Journal* and Performa computers this December. (See the “Apple Comarketing Opportunities” chart starting on page 29 for participation details on this and other programs mentioned in this article.)

Another recent organizational change that's already helping developers is the expansion of Apple Developer Relations and the formation of two new groups: Developer Marketing and Developer Business Development.

Chances are, you're already reaping some of the benefits of this change. For instance, Apple Developer News, which features new comarketing and distribution opportunities whenever they become available, is being delivered weekly rather than monthly. In addition, the “Apple Comarketing Opportunities” list (<http://www.devworld.apple.com/mkt/comarketing.shtml>), which is featured in this article, will be updated regularly at the Developer World Web site and in *Apple Directions*.

Last but not least, the transition of Apple's communication infrastructure from AppleLink to a Web-based information system is making Apple-related information more accessible to you and your customers, and, as is described in the next section, is fostering an array of new comarketing opportunities.

Please turn to page 31

Key Contacts for Comarketing Opportunities

As I stated in the body of the article, you need follow only two Apple-supplied electronic resources to keep informed about worldwide comarketing opportunities available to you. Here is the detailed information on how to access them.

- **The ADR Comarketing Web page.** Located at <http://www.devworld.apple.com/mkt/comarketing.shtml>, this Web page points you to detailed information on ongoing comarketing opportunities. You can access this information if you have any of the major Web browser applications and the ability to connect to the World Wide Web.
- **Apple Developer News (ADN).** This list server sends you developer-related news, delivered as e-mail, once a week. When late-breaking and one-time comarketing opportunities occur, Apple will let you know about them through Apple Developer News.

To subscribe to Apple Developer News, send e-mail to adirections@thing1.info.apple.com. In the subject field (not the message body), type the string “subscribe <your real name>”. ADN will then be automatically sent each Friday to the e-mail address that you used to send the “subscribe” message.

Here, in addition, are some other sites you might find of interest:

- **Apple Developer World.** This Web site, located at <http://www.devworld.apple.com/>, is Apple's main site for all types of developer support.
- **Market Information.** This Web site, at <http://www.devworld.apple.com/mkt/marketinfo.shtml/>, gives you information on the Macintosh installed base and various studies about the people who use Macintosh computers.
- **Developer Business Proposition.** This Web site, at <http://www.devworld.apple.com/con/appleprop.shtml/>, gives you information on why you should consider developing on various Apple platforms and how you can join any of Apple's developer-support programs.

Apple Comarketing Opportunities

Even if your company is outside the United States, U.S.-only opportunities apply to you if you market your product to the U.S.

General Opportunities

Apple Company Store

Enhance word-of-mouth marketing of your retail software products by displaying them at the Cupertino campus company retail store. You can also run Apple employee promotions through the store.

Cost: None

Deadline: Ongoing

Contact: Mark Wise, 408-974-8440, mwise@apple.com

Apple Developer News

Apple Developer News is the online business bulletin from Apple Computer that serves busy computer industry professionals. This weekly electronic newsletter is the best way to stay informed of Apple comarketing and distribution opportunities.

Subscribe by sending an e-mail message to adirections@thing1.info.apple.com. In the subject field, type the string "subscribe <your real name>".

Apple installed base access

Apple's customer database is accessible on a managed basis, providing you with a way to send product information directly to Apple's installed base.

Cost: Check the Web site for current costs

Deadline: Ongoing, available by October 1

Contact: Shirley McCoy, mccoy2@applelink.apple.com

Apple Mailbox

Include your product collateral (no disks) in this Apple corporate monthly mailing of sales tools, product materials, and program information. This mailing is sent to Apple field and corporate personnel, resellers, and other strategic external audiences.

Cost: None

Deadline: 19th of each month on a pilot basis

Contact: Shirley McCoy, mccoy2@applelink.apple.com

Apple Market Center seminars and events

Market Center events include seminars, product introductions, briefings, marketing campaign roll-outs, and solution days. Check out the seminar Web site to see how you can showcase your product during these solutions-oriented happenings.

Cost: None

Deadline: Ongoing

Contacts: *Publishing* — Susan Budner, 619-942-2622, budner.s@applelink.apple.com; *Science, Engineering, Design* — Christine Klimeczko, 215-557-3134, Klimeczko1@applelink.apple.com;

Communication/Collaboration — Todd Harper, 513-474-5951, harper@applelink.apple.com; *Small Business, Consumer* — Joanie Pond, 804-777-9723, pond1@applelink.apple.com
<http://www.seminars.apple.com>

APPLE CD demo space

This sales tool CD is shipped to more than 4,000 Apple sales and marketing employees and 25,000 Apple support providers, resellers, and VARs. Each quarter it features a variety of third-party applications, in order to display the breadth of Macintosh software available. Product demos on this CD are shown to thousands of Apple customers each year.

Cost: None

Deadline for the January 1997 edition: November 28, 1996; per-application space is limited to 56 MB

Contact: Apple Americas, Sales Communication, Susan Kind, susan.kind@apple.com

Australian Macintosh Product Guide

Include your product in this database of more than 14,000 commercial Mac OS-based software and hardware products. This guide is posted on the Apple Australia Web site, made accessible to retailers through a searchable database, shipped with Australia-based Performa computers, and printed for some Apple Australia events.

Target: Australian Macintosh users and retailers
Cost: None

Deadline: Ongoing

Contact: Apple Australia — Paul Netterfield, Barbara Willoughby, phone +61 2 456.48.45, cata@mpx.com.au

<http://www.apple.com.au/MPG>

Developer Central booth

At this time, Apple plans to make booth space available to developers in a special area at Macworld San Francisco in January and Macworld Boston in August.

Cost: Varies

Contact: Apple Developer Relations, Russ Havard, 408-974-1335, havard@applelink.apple.com

European Solutions Pavilions

Apple sponsors Mac OS Solutions Pavilions at key European events throughout the year. Participants can join forces with other developers to increase the awareness of third-party Mac OS solutions. As space is limited, Apple reviews the applicant pool and selects participants according to specific criteria for the event. Participants must have a product shipping in the region or language of the specific event within one month of the close of the event.

Cost: Apple and other Mac OS licensees who cosponsor the pavilion at a given location subsidize the cost of the space. Developers pay a participation fee of U.S. \$2,500 per event, and must provide sign content, PR materials, technical requirements, product copy, and a staff person.

Deadline: Ongoing

Contact: Jasmine Amirana, amirana@applelink.apple.com

It Shipped database

This rolling list of new and revised software product releases is published in the *Apple Directions* monthly newsletter and posted at the Apple Web site.

Cost: None

Deadline: For *Apple Directions*, by the 15th of every month

Contact: Apple Developer Relations, Gayle Ryan-Westbrook, 408-974-8939, westbrook@applelink.apple.com

<http://www.devworld.apple.com/mkt/itshipped.html>

Mac OS logo program



Benefit from Mac OS platform awareness by using the Mac OS logo artwork on your packaging materials and documentation. Apple will also be installing Mac

OS shelf talkers in retail outlets this holiday season.

Cost: None

Contact: Apple Software Licensing, 512-919-2645, sw.license@applelink.apple.com

<http://www.devworld.apple.com/mkt/registering/sw/swl.html>

Macworld Web Explorer Virtual Expo CD, cosponsored by Apple

Place product videos, demos, product information, and Web site links to this widely distributed virtual product expo CD. Apple is also placing its Macintosh Product Guide database on it.

Cost: Free registry product listing, virtual booths from \$3,600 to \$10,400

Deadline for the May 1997 CD: Full booth—November 1, 1996; other listings—November 30, 1996

Contact: For CD “booth”—Cherie LaFrance, 408-688-0707, cherie_lafrance@macworld.com

Picks of the Month

You can gain valuable publicity with 50 key press contacts if your product is chosen for this software seed mailing. Each month it features ten hot Macintosh products that are mentioned in a cover letter from Guy Kawasaki.

Cost: None

Deadline: Ongoing

Contact: Apple Developer Relations, Gayle Ryan-Westbrook, 408-974-8939, westbrook@applelink.apple.com

StartingLine sales and marketing tools (including the Apple Media Toolkit CD)

This catalog is Apple's distribution vehicle for marketing/communications, merchandising, exhibit, and collateral materials. Using these materials can save you marketing collateral costs. Through this catalog you can order the *Apple Media Toolkit* CD, which contains Apple product artwork, photographs, box shots, and lifestyle photography.

Cost: Inquire

Contact: 800-825-2145, merch.tsr@applelink.apple.com

Macintosh Software Guide

This soon-to-be expanded “product-finder” database will be available at the top level of the Apple Web site, and will provide users with links to developer Web sites.

Cost: None

Contact: Apple Developer Relations, Gayle Ryan-Westbrook 408-974-8939, westbrook@applelink.apple.com

<http://www.devworld.apple.com/mkt/thirdparty.html>

Consumer and Small Office/Home Markets**Family Computing Workshops**

This program is designed to introduce parents to the latest Mac OS computers and software for enhancing children's learning. Products featured in these workshops will reach more than 400,000 parents in 1996.

Cost: None

Deadline: Ongoing

Contact: Javier Villalobos, 408-862-6426, villalobos1@applelink.apple.com

Infomercial fulfillment product listing

A fulfillment packet is sent to people calling in about Apple's 30-minute infomercials. You can garner qualified customer leads by including your product listing in this kit.

Cost: None

Deadline: Ongoing

Contact: Javier Villalobos, 408-862-6426, villalobos1@applelink.apple.com

Internet Personalized Launcher

When a Performa purchaser fills out the new online registration card, a new program creates a customized Internet browser window for that user. Based on customer-supplied information, this window will provide Internet links to Apple and selected third-party Web sites, online technical support, and special customer offers.

Deadline: First day of each month

Cost: None

Contact: Javier Villalobos, 408-862-6426, villalobos1@applelink.apple.com

<http://myhome.apple.com/>

MacHome Journal Digest CD Sampler, cosponsored by Apple

This CD provides you with a great opportunity to gain exposure with home and SOHO users. It features software demos and will be shrink-wrapped with *MacHome Journal*. About 100,000 copies will ship in Performa boxes during the holidays.

Deadline for December issue: October 9, 1996

Cost: About \$400 to \$500 per MB

Contact for *MacHome Journal*—Susan Ford, 415-957-1911, ext. 14, susan_ford@machome.com;

Contact for Apple—Jack Rotolo, rotolo1@applelink.apple.com

Performa bundling

A few core third-party applications are shipped with every Performa, providing developers with

a high-volume selling opportunity. Apple looks for consumer solutions serving five markets: learning, entertainment, home productivity, Internet services, and home office. This opportunity is available only to a limited number of developers.

Deadline: Products selected from January to March

Contact for product review guidelines: Gene Longo, 408-862-7949, longo2@applelink.apple.com

Small Business Featured Developer

Each month a developer is featured in the Small Business area of Apple's Web site. See the Web site for participation details.

Cost: None

Deadline: Ongoing

<http://smallbiz.apple.com/featdev.html>

Enterprise, Scientific, Engineering, and Design Markets**Apple Business Bulletin**

This biweekly electronic newsletter is sent to large business users. It's filled with up-to-date news on Apple products, strategies, executive messages, promotional and program information, services, upcoming events, and related industry information. To spread the word about your large business product or service, send a press release or announcement by e-mail.

Cost: None

Deadline: Ongoing

Contact: bus.bulletin@applelink.apple.com

<http://www.solutions.apple.com/abb/abb.html>

Enterprise third-party solutions listing

This Web-based product listing can provide Apple customers with a description of your enterprise solution and a hot link to your Web site.

Cost: None

Deadline: Ongoing

<http://www.enterprise.apple.com/3rdparty.html>

SED featured developer

You can be considered for a free Science, Engineering, and Design (SED) Web page banner and hot link if your product serves the needs of the SED community.

Cost: None

Deadline: Ongoing

<http://www.technical.apple.com/>

Third-party solution of the month

You can be considered for a free Web page banner and hot link if your product serves the needs of Apple Enterprise, Science, Engineering, and Design markets.

Cost: None

Deadline: Ongoing

<http://www.enterprise.apple.com/entsol.html>

Education Markets**Apple Demo Days**

These events provide you with a way to demonstrate your education solutions to Apple evangelists.

Requirements: Developer must be an Education Solutions Provider (ESP) member

Cost: TBD

Contact: Check the *ESP Bulletin*, call 800-469-9523, or e-mail esp.program@applelink.apple.com

Apple Guide to Education Solutions

This searchable database will help customers locate Mac OS education-specific solutions, and will help you sell more of your products.

Cost: None

Deadline: Ongoing

Contact: Beverly Devilla, guides@applelink.apple.com, 408-738-4004, or esp.program@applelink.apple.com

<http://www.ednet.apple.com/resources/espprogram/espguide.html>

ESP bundling opportunities

A few core third-party applications are shipped with Apple systems sold to the various education markets. This distribution opportunity is available to a limited number of developers.

Deadline: Ongoing

Contact: 800-469-9523,

esp.program@applelink.apple.com

ESP Membership Directory

ESP's Web-based Membership Directory hot links can draw potential customers to your home page.

Requirements: ESP membership

Deadline: Ongoing

Contact: 800-469-9523,

esp.program@applelink.apple.com

<http://www.ednet.apple.com/Resources/ESPProgram/ESPMemberSearch.HTML>

Macintosh-only ESP database

This promotional customer database demonstrates the breadth of ESP software specifically designed for Mac OS platforms.

Cost: None

Deadline: Ongoing

Contact: 800-469-9523, esp.program@applelink.apple.com

ESP press kit inclusion and quotes

Periodically, third-party product information is included in Apple press kits. If you have a solution that shows off a new Apple technology, or if you want to share a success story for use in Apple sales materials and presentations, contact the ESP staff.

Cost: None

Deadline: Ongoing

Contact: 800-469-9523; esp.program@applelink.apple.com; obtain a quote and release forms from the *ESP Member Handbook*

Publishing, Entertainment, and New Media Markets**AMP Member Showcase**

The Apple Media Program (AMP) Member Showcase is a Web page featuring information on member products and services, and links to developer sites.

Requirements: AMP membership

Deadline: Ongoing

<http://www.amp.apple.com/>

Developer and New Technology Markets**Guide to Macintosh Development Tools**

If you market commercial, client/server, OpenDoc, Internet, or multimedia development tools, take advantage of this free product listing. It's posted on the Apple Tools Web page and distributed at the Developer Central Macworld Expo booth.

Cost: None

Deadline: September 15 and November 15, 1996

Contact: Apple Developer Tools, Mike Zivkovic, zivkovic@applelink.apple.com

<http://www.devtools.apple.com/general/guide2mactools/index.html>

Joint Solutions' Sampler CD and printed guides

This company's printed guides and sampler CD, which are distributed widely at Apple-related events and are posted on the Web, include information on commercial, Internet, Newton, and solutions development tools.

Cost: Varies depending on ad and memory space used

Contact: 408-471-1500, info@jointsolutions.com

Live Objects comarketing programs

Ongoing comarketing opportunities will arise as Apple steps up its Live Objects promotional efforts. Watch the Developer World page for participation notices.

Cost: None

CoMarketing oversight Contact: Stefan Schaefer schaefer@apple.com

<http://www.devworld.apple.com/mkt/comarketing.shtml/>

Direct Apple Customer Access

Over the years, Apple has offered a number of no-cost ways for you to leverage Apple mailings and events in promoting your product. The Apple Mailbox, Apple Market Center seminar collaborations, and the "Picks of the Month" PR mailings are perfect examples of this type of opportunity.

It's worth mentioning two other popular programs that have recently been reinstated at developers' requests. The "It Shipped" new product announcements are appearing in *Apple Directions* again, and are also being

posted at the Developer World site. Additionally, Apple is enabling developers to access Macintosh installed-base lists (derived from registration cards) for a nominal fee starting October 1, so you'll be able to mail product information directly to Apple customers. Watch the Developer World Web page (<http://www.devworld.apple.com/>) for upcoming information on this service.

Some Holiday Cheer— Finding Mac OS Software Will Be Easier at Christmas

Even though this is not a comarketing opportunity for you directly, we thought you should know: In addition to the customary print and television ads that Apple runs during the holidays, Apple is also working with key channel partners to draw more customers into software outlets. If your products are sold through the retail channel, this effort can result in incremental sales to you.

In an effort to combat the perception that

there are far fewer Mac OS titles than PC titles in software stores, Apple is working on two fronts. To raise the overall visibility of Mac OS software, Apple is paying to have Mac OS “shelf talkers” (little “happy Mac” shelf signs that protrude into the aisles) installed in key retail outlets. To help Mac OS users find compatible software, Apple is providing free “happy Mac” box stickers to developers who sell hybrid PC and Mac OS products. (By *hybrid*, we mean that the PC and Mac OS versions of the software product are shipped in one box. These hybrid boxes are often placed in the PC software section only.)

Leveraging the Web

Developers are already reaping the benefits of the Web at Apple’s many customer-targeted sites. Individual developers are featured with banners and links to their Web site from Apple’s enterprise, scientific, engineering, design, and education Web pages. Comprehensive product guides are posted for users looking for education solutions (Apple Guide to Education Solutions), developer tools (Guide to Macintosh Development Tools), or general Macintosh software (Macintosh Product Guide and Australian Macintosh Product Guide).

Still another innovative Web-related project, the Internet Personalized Launcher, is being initiated by Apple’s consumer group. This software program, which will ship with Performa computers, will provide you with a direct Web hot link to Internet-connected Performa customers. When Performa purchasers fill out an online registration card, the new Launcher will create a customized Internet browser window for that user. Based on customer-supplied information, this window will provide users with Internet links to Apple and selected third-

party developer Web sites, online technical support sites, and special customer offer sites.

Macworld and MacHome Journal CD Samplers

Developers who sell business or consumer software can also take advantage of two CD-ROM software samplers that are being cosponsored by Apple Developer Relations and the high-profile Macintosh publications, *Macworld* and *MacHome Journal*.

The *Macworld Web Explorer Virtual Expo CD*, set to debut in the magazine’s December 1996 issue, provides a good way for you to promote any type of mainstream Mac OS product. You can include a free product listing in the Macintosh Product Guide—a Web-based compatible product database maintained by Apple Computer—or you can buy demo space on the CD from *Macworld*. Depending on how much you want to spend, you can choose any of these three options:

- a full-level booth, with a custom-designed 90-second video of your product and company, a product demo, detailed product information, and a link from the *Macworld* CD Web site to your home page (special charter price: \$10,400, plus \$1,000 per link to your Web site)
- a junior booth, with a product demo, fact sheets, and a link to your Web site (special charter price: \$6,800, plus \$1,000 per link to your Web site)
- a virtual station, with product fact sheets, your logo, a photo of your product, and link to your Web site (special charter price: \$3,600, plus \$1,000 per link to your Web site)

The *Macworld* staff will perform all the creative work, including the video production,

voice-overs, and script writing, as well as the design of the selling environment on the CD. Though the deadline for purchasing space on the December 1996 CD has passed, there’s still plenty of time to submit materials for the May 1997 edition.

Apple is sponsoring a similar CD project with *MacHome Journal*, and it comes with an irresistible bonus: 100,000 surplus CDs will be printed and shipped with Performa computers during the holiday season. *You have until October 9, 1996, to submit materials for this great comarketing vehicle.*

A Word of Advice

As I said at the beginning of this article, the only two sources you need to keep track of to stay abreast of comarketing are the ADR Comarketing Web page (<http://www.devworld.apple.com/mkt/comarketing.shtml>), for ongoing opportunities, and our Apple Developer News listserver, for late-breaking and one-time opportunities.

Comarketing is an art, not a science, so be sure to pursue all the comarketing avenues that you think are relevant. If you do that, I’m confident that you will find one or more places where Apple can help you market your products to the buying public. ♣

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

[Author’s note: Many thanks to Kathy Sulgit-Johnson, Stefan Schaefer, Jack Rotolo, and Gayle Ryan-Westbrook for their help compiling the “Apple Comarketing Opportunities” chart.]

Starting with this issue, we’ll no longer include the Listings section in Apple Directions, although we’ll continue to update the information from it and post it to Apple’s Web site. See this month’s Editor’s Note on page 2 for more information.