



# AppleDirections

## Inside This Issue

Editor's Note: Momentum!	2
IndustryWatch: Demo '94	3
PowerPC News	9
Worldwide eWorld On-line Service Emphasizes Content, Cross-Platform Access	9
Apple Licenses DAL to Independence Technologies	10
Computer Telephony Expo to Be Held in March	11
Apple and Microsoft to Provide Cross-Platform Messaging and Collaboration Solutions	11
CD Highlights: <i>Little Bit Man</i>	12
Macintosh Computers With PowerPC Available in Third-Party Test Lab	12
PowerShare—The Workflow Genie	13
Secrets of Authentication Revealed!	14
Human Interface: Fluent Interfaces, Part One	18
The Story Behind the Power, Part Two	20
Market Research Monthly: RAM in the Global Installed Base	24
Marketing Feature 1: Your Ten Biggest Competitors, Part Two	26
Marketing Feature 2: Winning the Turf War	29
APDA Information	32

## Apple News

### Apple Licenses Bedrock From Symantec to Create OpenDoc Framework

On January 24, Apple Computer, Inc., and Symantec Corporation announced that Apple has licensed the development rights to Bedrock, a cross-platform application framework technology that both companies have been developing jointly. Symantec and Apple announced a strategic alliance to develop Bedrock in June of 1992.

Apple will use Bedrock technology to provide you with a framework that creates cross-platform OpenDoc parts. OpenDoc is an open, cross-platform compound document architecture designed by Apple and supported by companies including IBM, Novell, Sun, Taligent, and WordPerfect. (For more information on OpenDoc, see "Why 1994 Will Be Like 1984" in the August 1993 issue of *Apple Directions*.)

Using OpenDoc has important benefits for both you and your customers:

- You'll be able to develop products faster and with less development effort. Your customers will be able to custom-tailor their documents

*please turn to page 8*

## Strategy Mosaic

### A Phone on Your Desktop

**Your Macintosh Desktop, That Is**

*By Gregg Williams, Apple Directions Staff*

Apple doesn't just design add-ons, it designs *architectures*. And not just any architectures—architectures that last and provide low-level "plumbing," architectures that free you to be creative and do more than you could on your own. From the beginning, QuickDraw and the original printing architecture did a lot for you—and now we're improving both with QuickDraw GX. (Pity the poor DOS developer, who still has to maintain printer drivers for every printer on the market.) More recently, the Apple Open Collaboration Environment (embodied in Apple's PowerTalk and PowerShare products) gave you a flexible, implementation-independent way to send mail and messages and access directory and catalog information.

The Macintosh Telephony Architecture (MTA), which was introduced in 1991, makes telephony (the transmission of voice and data through various wired or wireless means) another technology that you can easily manipulate from within your application. You can begin to add telephony services to your application for any Macintosh computer running System 7 and connected to a modem. You can start out by simply implementing one Apple event in the Telephony suite, but you will be

*please turn to page 5*

# AppleDirections

Volume 2, Number 3

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

**Editor**

Paul Dreyfus (AppleLink: DREYFUS.P)

**Technical Editor**

Gregg Williams (GREGGW)

**Business & Marketing Editor**

Dee Kiamy (KIAMY)

**Designer**

Robert Stone

**Production Editor**

Lisa Ferdinandsen (LISAFERD)

**Contributors**

Rick Andrews, Juan Bettaglio, Pete Bickford, Pat Calderhead, Alex Doshier, Tim Enwall, Amanda Hixson, Regis McKenna, Geoff Moore, Stacy Moore, Kris Newby, Caroline Rose, Anne Szabla, Jessa Vartanian

**Manager, Developer Press**

Dennis Matthews

**Manager, Developer Support Information**

Greg Joswiak

**Production Manager**

Diane Wilcox

**PrePress/Film**

Aptos Post

**Printer**

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

© 1994 Apple Computer, Inc., 20525 Mariani Ave., Cupertino, CA 95014, 408-996-1010. All rights reserved.

Apple, the Apple logo, APDA, AppleLink, AppleShare, AppleTalk, EtherTalk, HyperCard, LaserWriter, MacApp, Macintosh, Macintosh Quadra, Newton, and PowerBook are trademarks of Apple Computer, Inc., registered in the U.S. and other countries. AOCE, Apple Desktop Bus, AppleScript, AppleSearch, DigiSign, DocViewer, eWorld, Finder, GeoPort, Macintosh Centris, MessagePad, OpenDoc, Performa, PlainTalk, PowerBook, PowerBook Duo, PowerShare, PowerTalk, QuickDraw, QuickTime, TrueType, and WorldScript are trademarks of Apple Computer, Inc. Adobe, Photoshop, and PostScript are trademarks of Adobe Systems Incorporated, which may be registered in certain jurisdictions. Classic is a registered trademark licensed to Apple Computer, Inc. DEC and VAX are trademarks of Digital Equipment Corporation. NuBus is a trademark of Texas Instruments. PowerPC is a trademark of International Business Machines Corporation, used under license therefrom. UNIX is a registered trademark of UNIX System Laboratories, Inc. X Window System is a trademark of the Massachusetts Institute of Technology. All other trademarks are the property of their respective owners.

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

**Editor's Note**

## Momentum!

It's in my job description as the editor of *Apple Directions* to convince you that Macintosh is a great platform and to make you feel great about developing new products for it. So, when I tell you that I think it's a fortunate time to be a Macintosh developer, of course you're going to take that analysis with a smallish grain of salt.

But this past month, two items crossed my desk that I think even Apple's critics would admit speak to the success of the Macintosh platform, both as a business and as a technology.

**Item 1**

In case you haven't heard the news, the market for your Macintosh products just grew by over 1 million computers.

According to its quarter-end report, Apple Computer, Inc., for the first time sold more than 1 million Macintosh computers in Q1 1994 (from October 1993 to December 1993, which Apple counts as the first fiscal quarter in 1994). This marks a 40 percent unit shipment increase over Q1 1993. In the United States, shipments through consumer channels more than *doubled* compared to the same quarter the year before. In Japan, unit shipments also increased *100 percent*; according to International Data Corporation, the Macintosh share of the personal computer market there grew to 14 percent from 8 percent in calendar year 1992.

I think that's phenomenal. Apple's remarkable sales took place at a time when many folks' "common sense," including some experts', suggested that demand for Macintosh computers would slacken as customers awaited the introduction of Macintosh with PowerPC. So much for common sense; I guess it's never really applied to the high-tech industry.

Also interesting to note is that Q1 1994 was Apple's second strong quarter in a row; Q4 1993 saw unit shipments increase 36 percent over Q4 of 1992. In Q4 1993, Apple also enjoyed revenues of \$2.14 billion, a 21 percent increase over the year before and a record for the company, one that lasted only one quarter, until Q1 1994 when revenues reached \$2.469 billion, 23 percent more than in Q1 1993.

**Item 2**

In his Personal Technology column in the January 13, 1994, *Wall Street Journal*, Walter Mossberg pays the Macintosh computer a highly visible compliment.

Writing about the "dirty little secrets of the [multimedia] industry, the dark side of multimedia," Mossberg says, "In many cases, getting multimedia hardware and, especially, software, to run on a typical IBM-compatible computer can be a frustrating—even impossible—task."

Note that he's talking about DOS/Windows multimedia, definitely not Apple's solution. Amid a list of problems with Brand X multimedia too lengthy to detail here, he points out, "In general, users of Apple's Macintosh computers face very few of these problems, because a single company—Apple—makes both the hardware and the operating system and has made sound, graphics, and video a priority."

In a subsequent telephone conversation, Mr. Mossberg told me the main point he was getting at was the integration of Macintosh software and hardware. All Macintosh users have to do, usually, is plug in a new hardware device or launch a new application, and the hardware or software works without requiring the kind of reconfiguration needed by many new products in the DOS/Windows, IBM PC-compatible world.

Getting back to the 1 million Macintosh computers sold last quarter, they're getting the Macintosh advantage, an advantage that the juggernaut from Redmond, Washington, has yet to overcome. As Mr. Mossberg confirms, the Macintosh computer is still easier to use; the other platform may be making noises about someday providing "plug and play," but the Macintosh has had it since day one, and Apple will continue to make its technology easier to use.

I'd say it's a great time to have Macintosh products on the market. We have more users and more momentum than ever—and the first Macintosh with PowerPC system hasn't even shipped!

*Paul Dreyfus*  
Editor

## IndustryWatch: News &amp; Perspective

## Demo 94

By Amanda Hixson, Director of Licensing,  
InfoWorld Publishing Company

I recently spent four days toughing it out at the Demo 94 conference in Indian Wells, California. Indian Wells, for those of you unfamiliar with the city, is one of those dense pockets of humanity stuffed between golf courses in the Palm Springs area to the east of Los Angeles, just outside the recent quake zone.

Demo is an annual computer conference hosted by David Coursey, editor of *PC Letter*, a publication owned by the International Data Group (IDG), which also publishes *Macworld*, *PC World*, *InfoWorld*, and *Computerworld* magazines, to name a few.

Demo's intent is to bring together industry movers and shakers with a limited number of hot new products and technologies in a friendly, hands-on environment. In contrast to most major trade shows, which overwhelm you with so many products and people that you can't really absorb anything, Demo shows only the best, and gives you plenty of time to explore each product in detail, discuss new ideas, and schmooze like crazy.

This year's Demo, the fourth such conference, began with a series of announcements that included a bevy of communication, workgroup, and on-line products. The nature of this year's announcements reflects a shift by the industry to distributed, information-intensive products and represents the industry's latest focus on distributed data and methods for accessing it.

The key new product announcements included Adaptive Software's PeopleScheduler; CrossTies, a personal information manager (PIM), and Face to Face, a collaboration product, both from CrossWise Corporation of Santa Cruz, California; WordPerfect's TeleMail e-mail and scheduling product; Gaia Personal Media, an interface for AT&T's EO Communicator, from Gaia of Portland, Oregon; and Ziff-Davis Interactive, an on-line service with a graphical user interface from Ziff-Davis in Cambridge, Massachusetts.

Stories about these products were carried in many major publications, so I won't go into a lot of detail about them; what you need to know about them is that they represent the industry's initial forays onto the information highway. Promoting himself last month, David Letterman likened his show to "the information superhighway without the information." These products, though, both extend the highway and provide information for its travelers; you can expect future products, including your own, to include features necessary for the long haul down the electron trail.

### Demo 94, Part 2: PowerPC

Not everything at Demo is new. The conference is also a great place for companies to update the industry on the latest developments in key technology and product areas. This year a critical update area was PowerPC technology. Representatives from the PowerPC troika—Apple, Motorola, and IBM—joined each other on the dais and brought us up to date on developments in this hot technology area.

Kudos must go to Jim Gable, Apple's PowerPC product line manager, who performed absolute magic in a demo of Macintosh with PowerPC technology. PowerPC is the peg on which Apple is hanging its future, and Jim's demo made me think that the peg can carry a heavy load.

I was dazzled as I witnessed several key applications, including Microsoft Excel, running in native mode at speeds up to 80 MHz. It was even more amazing to see Insignia Solutions' SoftWindows, a Windows emulation program, running Windows applications at 80486 speeds on a PowerPC processor-based Macintosh computer.

Based on what I saw at Demo, the new RISC Macintosh computers will definitely offer customers the best of both worlds, with great performance. If you're not developing native applications for Macintosh with PowerPC, you ought to be. As developers, you should be doing all you can to ship as many PowerPC applications as quickly as possible. If you support Macintosh with PowerPC, Apple and you together will be able to attract many new customers.

### Demo 94, Part 3: Newton

I recently became the owner of a Newton MessagePad, and since then, I've become convinced that what will make customers take Newton across the new technology product chasm is a compelling application. That's where you guys come in.

At Demo, Philip Ivanier, manager of developer relations for Newton, showed a half-dozen Newton applications. The applications included a program designed to help golfers improve their games, a Tupperware sales tool, and a really cool expense-management package that is intelligent enough to know that "Pizza \$28.50" is probably related to a meal, such as lunch or dinner.

It was apparent to me during the presentations, and to Dave Coursey, with whom I discussed this afterward, that I was looking at some pretty terrific applications. Unfortunately, when put to a vote of the audience during the conference, none of them was compelling enough to convince a majority of the folks in attendance that they need to own a Newton device.

## Apple Directions On Line—April

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

March 1—Preliminary draft copy

March 15—Final copy

To view the April issue of *Apple Directions* on line, follow the AppleLink path Developer Support:Developer Services:Periodicals:Apple Directions:Apple Directions April 1994.

Just as VisiCalc (I know, I'm dating myself) made the Apple II and the early IBM PC worth owning and Lotus 1-2-3 carried the PC into puberty, the Newton device needs a killer application to make it a must-have product. This is an opportunity for those of you who have not yet decided whether to develop for Newton. Newton, as a technology and as an Apple product, is going to be around for awhile. One of *you*, though, has the idea that will move Newton into its next phase of market acceptance.

#### Demo 94, Part 4: Dirty Windows

I know I shouldn't get into this, but I'm going to anyway.

One of the things shown at Demo was a preview of the new interface for Chicago, an upcoming version of Windows from Microsoft due at the end of this year. I gotta tell ya, when the person doing the demo opened a window filled with folder icons to a round of applause, I couldn't believe it. You've got to understand, these icons were the type that appeared on the Macintosh computer *ten years ago*. Making matters worse, filenames longer than the traditional 8.3 found on current DOS/Windows machines (filename.gud, for example) were treated as a revelation, as if Microsoft had invented the idea.

You'd be doing yourselves and your customers a favor if you helped spread the word that the Macintosh computer has had features as obvious as a hierarchical file structure and long filenames (not to mention plug-and-play capability and the Finder) since before Windows was a gleam in Microsoft's eye. While you're at it, you might also say that Apple is continuing to advance the Macintosh interface while others are still just trying to catch up to where it was in 1984.

#### Demo 94, Part 5: Macintosh Developer and Evangelist

It's not often that I see anyone who harks back to the old days of Apple, but there's a guy out there whose demos are worth watching just by virtue of their infectious enthusiasm. I'm referring to Kai Krause, vice president of HSC Software. HSC produces Kai's Power Tools, a set of plug-ins for Adobe™ Photoshop.

I already own both Kai's Power Tools and Photoshop. (I don't do any serious work with either product, but I like to play with them to amuse myself on weekends.) After watching Kai demonstrate his power tools on a standard Macintosh Quadra computer and then again on a Macintosh computer with PowerPC, I was ready to go out and buy the product again. Both Kai and his product are that good.

Even if you can't make it to a Macworld or some other venue where Kai shows his wares, you can contact Kai on a number of on-line services. He has a ton of terrific ideas, and it would be well worth your time to chat with him on-line if you get the opportunity. Personally, I give Kai the Amanda Hixson "1984" award for enthusiasm and evangelism in the tradition of the early Macintosh pioneers.

---

*Amanda Hixson, long-time Apple veteran, high-tech journalist, and industry analyst, is currently director of licensing for InfoWorld magazine. There, she's responsible for discovering new forms of electronic distribution for the magazine's content. Don't worry, she isn't responsible for creating the content, simply making sure that all of you have on-line access to it. She can be reached at A.HIXSON on AppleLink.*

PowerPC™ Technology  
 PowerPC Technology  
 PowerPC Technology  
 Coming Soon to Macintosh  
 PowerPC Technology  
 PowerPC Technology



## Strategy Mosaic

## A Phone on Your Desktop

continued from page 1

able to do much more if you implement the entire suite.

But the real win comes when you integrate the Macintosh Telephony Architecture with other key Macintosh technologies—such as AppleScript, PowerTalk, PowerShare, and PlainTalk speech-recognition and text-to-speech—to create sophisticated business solutions impractical until now. (For one interesting set of possibilities, see the text box “No-Hands Telephony and the ‘Invisible Assistant.’”)

Here’s another way to look at it: How many computer users *don’t* have a telephone nearby? Not many. If you can come up with a “killer application” that allows a Macintosh and a telephone to do much more together than either one can do by itself, you will have one large and eager audience.

### Why MTA Will Succeed

The Macintosh Telephony Architecture is a good solution because it allows you to tap into telephony services with maximum effect and minimal effort. This happens in two ways:

- You can add telephony support with only a modest amount of coding.
- You don’t have to change your code if the user changes the underlying telephony service.

The Macintosh Telephony Architecture achieves these effects through *abstraction*, a technique by which applications can interact with a standard representation of a telephone, and each method of connection is made to act like the standard telephone that the application knows how to manipulate. (More on this later.)

OK, we’ve all seen good solutions that aren’t commercially successful; why will the Macintosh Telephony Architecture be any different? I think it will succeed, in addition to the reasons given earlier, because it brings a critical mass of both technical and market-driven advantages:

- Your program will have more great features because you won’t have to build the necessary

infrastructure; Apple’s done it for you.

- This architecture will create new markets for you. In addition to new solution products for customers, you may also be interested in creating some of the building blocks that make up the Macintosh Telephony Architecture. (This includes both hardware and software products.)

- The Macintosh Telephony Architecture is “blessed” by Apple Computer, Inc., has a strong feature set, and is open to all developers. Because of this, telephony-aware applications will be compatible with each other and, together, will be more useful to Macintosh owners.

- Finally, the compatibility and synergy of MTA-based products will result in a wider acceptance of telephony and a larger,

## No-Hands Telephony and “Invisible Assistant”

Jabra Corporation is about to prove two important things: one, that a small company with a vision can still potentially revolutionize the Macintosh industry; and two, that the Macintosh system-level technologies are the building blocks that make such revolutions possible.

Jabra, a company based in San Diego, California, has a new product out for the Macintosh Quadra 660AV and 840AV computers, the Jabra Ear Phone Streamline AV. This product, when placed in your ear, serves as both a speaker and microphone and allows you to conduct a telephone conversation without using a handset or headset. Not only is the operation hands-free, but what the other person says is (literally) for your ears only.

### “Take-It-With-You” Computing Power

But that’s not where the vision comes in. I recently talked with Randy Granovetter, president and CEO of Jabra, and David Roach, Jabra’s vice president of research and development. According to them, sometime in the first calendar quarter of 1995, Jabra plans to ship a wireless version of their Ear Phone Streamline

AV that will have a range of several hundred feet.

Now this makes things *interesting*. Making a critical phone call while you’re in a meeting or in someone else’s office is just the start. Consider what you might do when you add key Apple technologies such as PlainTalk speech recognition and text-to-speech, GeoPort, scripting, PowerTalk, applications that communicate through Apple events, AppleTalk networking and file sharing, and AppleSearch.

If Jabra’s wireless Ear Phone Streamline AV product lives up to its promise, it will be part of an entirely new paradigm for computing: voice-driven computing. It will be possible to create, in essence, an “invisible assistant” that goes wherever you do. No keyboard, no electrical plug, no video display, no bulky objects to carry around—you talk, and this “assistant” does your bidding.

### Things to Come

It’s not too “blue sky” to imagine a Macintosh computer responding to the following voice commands:

- “Computer, fax the document named *Quarterly Report 93* to Joe in Dallas.”
- “Computer, set up a conference call with Julie and Mike.”
- “Computer, are there any documents on the networks that contain the words *auto, sales, and 1993?*”
- “Computer, read all new e-mail.”
- “Computer, schedule a staff meeting for Thursday at 10:00 and tell me who can’t attend.”
- “Computer, open the file *Projected Sales*. Change the field *Profit Margin* to 20. What is the value of the field *Total Sales?*”
- “Computer, call home. Repeat until you get an answer. Connect me to the call.”

If these examples don’t spark your imagination and give you a glimpse of entirely new products and markets, you’re in the wrong business. And remember, all these Apple technologies will be available on every PowerPC processor-based Macintosh Apple sells, and Apple plans to be selling a lot of them—soon.

healthier market for telephony-based products.

### How MTA Works

Though the Telephone Manager is at the heart of the Macintosh Telephony Architecture, most of you won't deal with it directly. As shown in "The Macintosh Telephony Architecture" (below), the Telephone Manager provides low-level control of telephony services.

First, some terminology: The Telephone Manager works with a telephone tool to make a given telephony service available to clients of the Macintosh Telephony Architecture. Telephony service can be accessed directly through a phone line coming to a user's desk (this is called *first-*

*party call control*) or through a server connected to a private branch exchange, or PBX (this is called *third-party call control*).

When the connection is to the phone line at a user's desk, the telephony adapter provides the physical interface. If the adapter is capable of providing functions in addition to signaling, an adapter handler, along with other tools, drivers, or components, provides access to these other functions through the appropriate Macintosh APIs. Examples of telephony adapters include Apple Desktop Bus (ADB) devices that dial phone numbers, modems connected to serial ports, NuBus™ cards (such as ISDN cards), and, of course, GeoPort adapters (more on this later).

At the top of the Macintosh Telephony Architecture is the telephony-aware application, which can be any application that can make use of telephony services. This application communicates with the layer below it solely through the Telephony suite of Apple events, which allows applications to interact with telephony services without needing to know implementation-specific details.

Note that any application that is said to be "AppleScript-attachable" (meaning that either you or the user can add scripts to user-interface elements like menus and buttons) can be made telephony-aware by adding the right scripts. Though this is a good short-term solution, you should still plan to integrate telephony Apple events directly into your application.

The Telephony suite includes Apple events that allow an application to do things like make, answer, hold, park, and redirect calls. Through a mechanism called a *dependency*, an application can require (through a Notify Dependency Apple event) that it be notified when certain telephony events occur. (For example, an application could be notified when a phone call ends and could log the person called and the duration of the call.)

### Below the Telephony-Aware Application

Telephony-aware applications send telephony Apple events to one of two types of applications, screen-based telephony applications or programmed telephony applications, which then interact with the Telephone Manager. Why this extra layer? The user interface. The Macintosh Telephony Architecture separates function (telephony events controlled by the telephony-aware application) from the user interface (what the user sees)—a design decision that makes your application design more flexible and easier to implement.

This extra layer provides the most appropriate user interface for both the computer being used and the function being performed. For example, a desktop Macintosh user may need a different interface from, say, a Macintosh PowerBook user, whose screen space is at a premium; and the interface for an application used primarily for personal dialing will be different from one used primarily to coordinate conference calls. A Macintosh computer will use only one such application—after all, it presents the user interface elements on the screen that represent the telephone—and it will interact with every telephony-aware application (with telephony Apple events traveling in both directions).

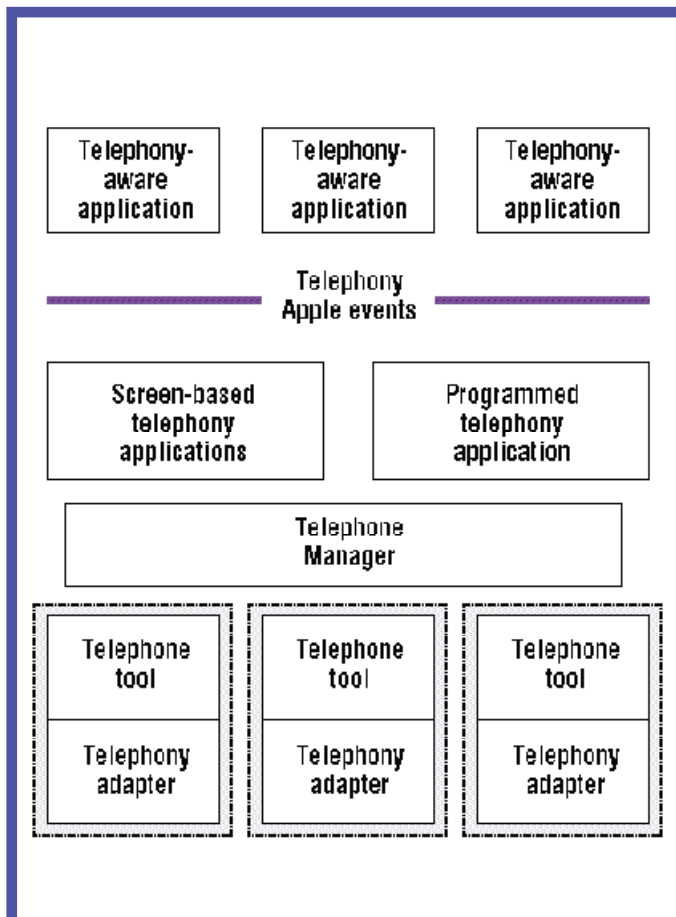
Most of you will simply use telephony Apple events to add telephony to your applications; you'll assume that the user will supply the appropriate screen-based or programmed telephony application. However, in some situations, you may wish to supply a screen-based or programmed telephony application with your telephony-aware application, or you may find a market in providing one of these two applications to users.

Because people will use screen-based telephony applications and programmed telephony applications in different situations, I'll discuss them separately.

### Screen-Based Telephony Applications

These applications primarily bring Macintosh integration and ease of use to telephone activities and create a user interface that puts a virtual telephone on the user's Macintosh desktop. Since most people interact with their telephone frequently (if not constantly), screen-based telephony applications will definitely have a large target audience.

A screen-based telephony application, depending on the



**The Macintosh Telephony Architecture. Most developers will create telephony-aware applications, which use telephony Apple events to add telephony functions. The layers underneath shield the applications on top from needing to know implementation details.**

user's needs, may do things like the following:

- display the status of a call
- transfer and forward calls
- dial a number and provide visual feedback on the Macintosh desktop
- display the caller's name and phone number before the call is answered
- maintain a log of incoming and outgoing calls and their durations for billing purposes
- serve as a visual switchboard for anyone handling the calls of a group of people

Here are some examples of telephony-aware applications that would make use of a screen-based telephony application:

- a database (or any other application) that can call any phone number in a record through a simple menu command, button, or keystroke
- a client-tracking system that can automatically display clients' records when they call (this requires a phone that has the caller ID option)
- a personal information manager (PIM) that reminds users of scheduled telephone calls and places the calls automatically
- a conference application that sets up conference calls and allows participants to share information while the call is taking place

### Programmed Telephony Applications

These applications differ from screen-based telephony applications in that they may not interact with a telephony-aware application. (They may, however, interact through Apple events with other applications to trigger actions or retrieve information.) In fact, the whole idea of a programmed telephony application is that it provides some service automatically, without human intervention. Some examples are

- voice mail systems
- automated receptionists
- automated paging and call-routing systems
- telephony-based information retrieval systems

If you use the PlainTalk text-to-speech capability, which is available for medium- to high-end Macintosh computers, you can do things like retrieve your electronic mail from a remote phone or provide customers with virtually unlimited amounts of "canned" information (stored in text files and read by PlainTalk) from any touch-tone phone.

For Macintosh AV computers and the upcoming PowerPC processor-based Macintosh computers, you will also have both PlainTalk text-to-speech and speech recognition, which will make even more powerful features possible.

### GeoPort

Just as AppleTalk made local-area networking simple and inexpensive, Apple's GeoPort technology does the same for wide-area communications. GeoPort can connect analog or digital phone lines to any personal computer, personal digital assistant (PDA), or workstation, connecting through a standard serial interface but transferring data over 100 times faster than usual. Apple intends to make GeoPort a platform-independent standard, which it feels will promote the integration of telephony into personal computers and give consumers a wider range of products to choose from.

In essence, GeoPort technology comes in three parts:

- the GeoPort adapter, which provides the electrical, physical, and logical connection to a telephone line or lines (depending on its design, a GeoPort adapter may support either a single connection or multiple connections simultaneously)

• the GeoPort communication protocols, which provide the initialization, setup, and data-streaming conventions for both the adapter and the computer it's connected to (these protocols can handle dozens of data, voice, and signaling streams)

- a GeoPort serial driver, which runs on the computer and connects the various data streams to the appropriate hardware or software

Apple currently sells a GeoPort adapter that connects a Macintosh AV computer to a basic analog telephone line. Apple is working

to make available additional GeoPort adapters that connect to ISDN and digital PBX services. If you have experience in telephony, you may be interested in creating a GeoPort adapter to fit a specific telephony technology.

### PowerTalk

You don't have to have PowerTalk to get started with telephony, but your telephony-aware application can do more impressive things if PowerTalk is present. The main contribution of PowerTalk is its catalogs and information cards, which include

## Resources

*Telephony Apple Events Suite.* Needed to create telephony-aware applications. Located on the December 1993 Developer CD, pathname Dev.CD Dec 93:Reference Library:Technical Documentation:Apple Events:AE Suite • Telephony.

Telephone Manager 1.1.1 (extension). Software is available on the February 1994 Developer CD, pathname Dev.CD Feb 94:New System Software Extensions:Telephone Manager 1.1.1. The software with documentation is available in the Telephone Manager Developer's Kit, available from APDA (#R0145LL/A, \$75.00 in U.S.). You need the documentation only if you want to create screen-based telephony applications or programmed telephony applications.

*Macintosh Developer Note #5: Macintosh Quadra 840Av and Centris 660Av Computers.* Part Three of the book documents the development model for PlainTalk speech recognition and text-to-speech. Available from APDA (#R0529LL/A, \$60.00 in U.S.). The Macintosh Centris 660Av was later renamed the Macintosh Quadra 660Av.

*Inside Macintosh: Interapplication Communication,* published by Addison-Wesley. Explains how to add Apple events to your application. Available from bookstores or from APDA (#T0594LL/A, \$36.95 in U.S.).

Apple Events/AppleScript Programming Tutorial. Self-paced tutorial that teaches you how to add Apple event and AppleScript support to your application. Available from APDA (#R0224LL/A, \$150.00 in U.S.).

AppleScript Software Development Toolkit version 1.1. Needed to add AppleScript support to your application. Available from APDA (#R0175Z/B, \$199.00 in U.S.).

*Note: See page 32 of this issue for information on contacting APDA.*

the phone numbers for the people with whom a user wants to collaborate. In addition, the PowerTalk mailbox is the place where incoming voice mail messages arrive, and when your mail-capable application uses PowerTalk to send an item using a telecommunications-based mail service, PowerTalk establishes a connection using the appropriate mail server.

### Apple Telephony Starter Kit

Apple wants to make it as easy as possible for you to get started adding telephony support to your application. An Apple Telephony Starter Kit, to be available sometime this spring, will give you the basic tools you need to start—all you have to do is add telephony Apple events to your application. (*Apple Directions* will give you further details on the kit's availability as soon as possible.) This kit will include

- a basic screen-based telephony application that is Apple-

Script-recordable (your application will send telephony Apple events to this application)

- a basic telephone tool that works with any Hayes-compatible modem
- another telephone tool that works with Sophisticated Circuits' Desktop Dialer (an ADB hardware add-on that dials a phone number for you)
- sample AppleScript scripts that use telephony Apple events

### So How Do I Start?

If your application uses phone numbers in any way, you should at least make your application able to dial phone numbers for the user. To do this, you need only implement the Telephony suite's Make Call Apple event and its standard representation of phone numbers (the cPhoneNumber object class). But if you implement the entire Telephony suite (twelve Apple events and five object classes), you will be able to do much more with telephony, both now and later.

You should also convert your application to store every phone number in the cPhoneNumber form, which contains information that applications will need—the country code, for example—to make a call from any part of the world. Users are becoming increasingly mobile, and they will appreciate a program that dials phone numbers correctly wherever they are. They will also appreciate a program that takes a phone number, however they enter it, and stores it in the correct form without forcing them to enter every detail. (For example, an address-book program should look up an entry's address and fill in the country code automatically.)

Beyond this first step, you should think about new products and enhancements to existing products. *Everybody* uses telephones, and there must be hundreds of ways that a Macintosh computer tightly integrated to a telephone or a telephony service can make people more productive and efficient. Telephony can enhance generic, horizontal

applications, but it can also be used to create powerful (vertical) solutions to specialized problems.

You may also want to create other pieces within the framework of the Macintosh Telephony Architecture itself. This includes screen-based telephony applications, programmed telephony applications, and hardware/software connections to specific telephony services.

### Put the Pieces Together

Remember, however much you can do with the Macintosh Telephony Architecture, you'll be able to do much, much more if you use the other technologies that the Macintosh platform makes available—things such as speech recognition and text-to-speech, programs working together through Apple events and scripting, simple and powerful networking, and integrated communications through PowerTalk and GeoPort. It's going to be exciting. ♣

## Apple News

### Apple Licenses Bedrock

*continued from page 1*

by using their choice of OpenDoc parts, and they'll be able to exchange documents with users on other platforms.

- You'll be able to update or enhance your applications more easily, because you'll only have to change individual parts instead of the whole application.
- You'll be able to create more bundles that can be targeted at vertical groups or specific user types. This will also be an advantage to your customers, since they'll get exactly what they need from their software.

Apple decided to redirect the Bedrock project because of the advances in software technology (like OpenDoc) that have been made since the project's start in mid-1992. Along with other companies, Apple has decided that the OpenDoc compound-document architecture is key to creating more powerful software in the future. Apple concluded that it was critical to integrate OpenDoc into Bedrock now so that developers will have a tool they can use to create cross-platform OpenDoc applications. Given that Apple designed OpenDoc and can more tightly integrate it into Bedrock than Symantec can, both companies reached an amicable agreement that

allowed Apple to license the rights to the Bedrock technology.

Since frameworks are designed to make it easy to implement the basic behaviors that are common to most software, OpenDoc will make it easier for you to implement standard Macintosh behaviors (such as Macintosh Drag and Drop) and key OpenDoc operations (for example, how to make an OpenDoc "container" and how to embed one OpenDoc part within another). By using OpenDoc, Apple will be able to develop and deliver new technologies in a form that will be easier for you to integrate into your applications.

### Symantec's Future Directions

Under the terms of their joint agreement, both Symantec and Apple have the right to develop Bedrock further, although only Apple has rights to commercially market the stand-alone product. Symantec will continue to use Bedrock internally, and Symantec's feedback to Apple will help guide Bedrock's evolution.

Symantec has made the business decision to develop tools for Microsoft's OLE 2.0 because that technology is on the market today. However, this does not detract from the future of OpenDoc. Technology under development in the Component Integration Laboratories (which administers the OpenDoc



standard) will enable you to take OLE 2.0 parts and use them in an OpenDoc environment, as well as to take OpenDoc parts and use them with OLE 2.0.

Symantec and Apple retain a good relationship and will continue to work together. Symantec is currently working with Apple on development tools for the PowerPC processor-based Macintosh.

Apple is establishing a timetable for delivering OpenDoc to Macintosh and Windows developers, and *Apple Directions* will keep you informed of details as they become available. Until then, you can contact the OpenDoc team at AppleLink address OPENDOC if you are interested in OpenDoc development and would like to receive an early seed version. To help the team in their final selection of seed sites, please describe your plans for using OpenDoc.

---

## **PowerPC News: Bundling With SoftWindows, More Upgrades, and Japanese Apps**

Many of the soon-to-be introduced Macintosh computers with PowerPC will be able to run Windows applications out of the box. That's the result of a recent agreement between Apple Computer, Inc., and Insignia Solutions to bundle Insignia's SoftWindows with selected PowerPC processor-based Macintosh systems.

In related news, Apple said that it will offer PowerPC upgrades for additional 680x0-based Macintosh computers and announced that 26 developers are

developing native PowerPC applications for Japan, currently the fastest growing geographic market for the Macintosh computer.

### **PowerPC to Do Windows**

In addition to being compatible with virtually all current 680x0-based Macintosh applications, Macintosh with PowerPC will also be compatible with DOS/Windows applications, thanks to SoftWindows.

In the words of Apple President Michael Spindler, "Just as we have focused on excellent compatibility with current Macintosh applications for Macintosh with PowerPC, Apple wants to provide multiplatform customers with excellent options for accessing thousands of existing applications. SoftWindows will give DOS and Windows users an easy path to the powerful new computing platform—and the promise of hundreds of new applications now under development that will harness the full benefits of PowerPC technology."

SoftWindows software lets Macintosh with PowerPC users run DOS and Windows applications at 80386 to 80486 performance levels, depending on system configuration. Apple plans to offer specific Macintosh with PowerPC configurations with SoftWindows preinstalled when it debuts the new computers in the first half of 1994. The product will also be available separately throughout the world.

Insignia's SoftWindows is an important part of Apple's marketing strategy for the new RISC-based Macintosh systems; it's intended to expand the market for Macintosh products by attracting customers who've previously purchased Intel 80x86-based PCs that run DOS and Windows.

As Chris LeTocq, director of software research at InfoCorp, points out, SoftWindows "provides both Insignia and Apple with an excellent opportunity to

address a much wider audience." (See "The Story Behind the Power, Part Two" on page 20 for more about Apple's efforts to attract DOS/Windows "switchers.")

SoftWindows provides full DOS and Windows in standard mode, as well as built-in PC network support. SoftWindows is also compatible with the full range of PC devices and systems, including COM and LPT ports, floppy disk drives, memory systems, video displays, networks, and CD-ROM drives.

### **PowerPC Upgrades**

Apple is also answering another compatibility question with its plans to make PowerPC hardware upgrades available for virtually the entire current 680x0-based Macintosh product line.

Recently, Apple announced it will provide upgrades for its current line of all-in-one Macintosh computers, including the Macintosh LC 520, 550, and 575, as well as the Macintosh Performa 550 computers.

In addition, Apple plans to make future PowerPC upgrades available for Macintosh LC 475, Quadra 605, and Performa 475/76 computers. The upgrades will boost performance two to four times on existing Macintosh models when running native applications.

These upgrade plans are in addition to those previously announced, which include upgrades for Macintosh Quadra 840AV, 800, 660AV, 650, and 610 models; the Macintosh Centris 660AV, 650, and 610 computers; and the Macintosh IIvx, IIvi, and Performa 600 products. Additionally, owners of the Apple Workgroup Server 60, 80, and 95 will be offered logic board upgrades to PowerPC processor-based systems, which will run a version of the Macintosh System 7 operating system for the PowerPC processor.

### **Japanese PowerPC Applications**

Finally, thanks to the efforts of many of you, Macintosh with PowerPC is poised to add to Apple's current success in Japan, where Macintosh unit shipments increased by 100 percent in 1993.

Twenty-six developers have announced plans to deliver new Macintosh with PowerPC versions of their products for the Japanese market, which will give their customers access to the advanced performance and features of the new platform.

The Japanese developers are Aldus Corporation, Inc., SRA (ACI), ErgoSoft, Catena (Language Engineering), Koshin Graphic Systems, Something Good, CRC Systems, SystemSoft, Justsystems, Subaru International (Specular), Seiwa Systems, Software Too (Quark), Dynaware, Polaroid Japan (Delta, B&E Software), WordPerfect Japan, B U G, HewLinks, Focal Point Computers, Mercury Software, Marubeni Electronics (Daystar), Maytrix (Dantz Development), LetraSet Japan (Fractal Design, VideoFusion), Works Zebra (Virtus), Graphisoft Software Development (USA), Wolfram Research (USA), and Deneba Software (USA).

---

## **Worldwide eWorld On-Line Service Emphasizes Content, Cross- Platform Access**

Apple Computer, Inc., recently announced eWorld, a new family of on-line services that will first be available for the Macintosh platform in the United States. The eWorld service will eventually be

available worldwide (in different languages) and on multiple computer platforms (Macintosh, Newton, and Microsoft Windows). Apple On-line Services (AOS) will also be providing tools that will help developers and information providers offer new products and services on eWorld.

Recognizing the appeal of the familiar, eWorld is modeled on a real-world metaphor, presenting people with a bird's eye view of an on-line community. The eWorld community consists of an electronic neighborhood of "buildings," each representing a specific area of the on-line service—the Library for research, the Newsstand for news and sports publications, the Business and Professional Plaza for business information and services, the Arts and Leisure Pavilion for entertainment and hobbies, the Computer Center for computer assistance and software, the Marketplace for purchasing products and services, the eMail Center for worldwide electronic mail (including Internet access), and the Community Center for interactive communications ("chat" and discussion areas) and on-line events. This extension of the real-world metaphor is supported by color-coded organizational schemes, icons representing standard eWorld functions, and sound to provide useful cues and helpful feedback.

eWorld is uniquely designed to be a global on-line service. It includes support for multiple languages for both content and applications, a global/local content model that allows publishers of all sizes to reach a global market and still offer information of local interest, and network services from multiple vendors providing local access points around the world. eWorld will premiere in the United States, but English-language versions of eWorld for Macintosh will be extended to countries around the world in

1994, followed by native-language versions for German, Japanese, and French.

In the future, eWorld services will be made available on a range of devices, including Macintosh personal computers, Windows PCs, and Newton devices, and people will be able to access common features across the different platforms. Services and information will also be consistent across supported platforms, allowing Macintosh and Windows users to interact in forums, post messages to bulletin boards, send mail, and perform transactions.

### **Publishing Tools**

Publishers will find eWorld an appealing environment, and developers may find business opportunities in helping content providers bring their data to the eWorld environment. Apple On-line Services has designed powerful publishing tools to simplify the creation and maintenance of on-line publications. Under the name *eWorld Press*, these tools will allow publishers to design and prototype new on-line products. Publishers can then use these tools to update those products in a cost-effective way by moving information from their existing repositories to eWorld's global servers and on-line services infrastructure.

Over 70 publishers and information providers intend to make electronic publications and services available in eWorld. Companies that have announced their intent to publish on eWorld include the Boston Computer Society (BCS), Berkeley Macintosh Users Group (BMUG), Claris Corporation, Dow Jones Business Information Services, Grolier Electronic Publishing, Inc., INDIVIDUAL, Inc., Regis McKenna, Inc., Reuters America, Inc., Tribune Media Services, USA TODAY Information Center, and WordPerfect Corporation. Magazines and magazine services that

will publish using eWorld include *Inc.*, *InfoWorld*, *Macworld*, and *ZiffNet/Mac*.

These companies will provide services such as user-group and product support, computer-product buying information, late-breaking financial and market news, personalized daily news services, on-line versions and enhancements of existing printed magazines and newspapers, and databases of consumer information (movie, book, and music reviews, television listings, and health and financial information).

### **Pricing**

In the United States, the basic monthly subscription fee for eWorld will be \$8.95, which will include two free hours of evening or weekend use. Each subsequent hour of evening or weekend use will be \$4.95. An additional network surcharge of \$2.95 per hour will apply during business hours in the United States. There will be no surcharge for either 9600 baud access or use of the Internet mail gateway. Future Macintosh customers who receive the eWorld software bundled with new Macintosh computers (which will begin by the end of 1994) will not pay a sign-up fee. Pricing for services outside the United States will be announced later.

---

## **Apple Licenses DAL to Independence Technologies**

If you're a Data Access Language developer, from now on you'll have to turn to a different source for support and new versions of the product. That source is Independence Technologies, Inc. (ITI), a Fremont, California,

networking developer that recently licensed DAL from Apple Computer, Inc. ITI intends to enhance and expand DAL technology as well as broadening service and support to those of you who use DAL.

Under the agreement with Apple, ITI will take over all current DAL licensing and support agreements as well as relations with DAL developers and resellers. In addition, ITI will market DAL under its own label, although DAL will remain available worldwide through APDA, except in Europe where it will be sold by Apple.

If you're used to turning to Apple for DAL support, however, you'll now need to call ITI Customer Support, unless you're in Europe, where you can continue to call Apple for the time being. ITI has announced a new family of DAL support services that will provide greater support than has previously been available. You can call ITI Customer Support at 510-438-2095.

ITI's mission will be to develop DAL technology beyond today's version 1.4. ITI will deliver DAL upgrades and extensions previously announced by Apple. Apple's statement of direction for DAL 2.0 will be fulfilled by ITI in 1994 when ITI ships the next generation of DAL. Most of Apple's announced plans to enhance cross-platform application development between DAL and Open Database Connectivity (ODBC) will be carried out by ITI in the first half of 1994, although Apple will be continuing development of the ODBC Driver Manager API for Macintosh.

Based on the Structured Query Language (SQL), DAL is a data access product that provides standard desktop applications with access to the best-selling databases. The DAL client resides on computers running the Macintosh or Windows operating systems. The DAL Server

provides data access to nearly 20 different relational databases, with more than 40 applications and tools currently available. A single application programming interface (API) allows programmers using the DAL Developer's Toolkit to create mixed-environment client-server applications with DAL.

---

## Computer Telephony Expo to Be Held in March

If this month's Strategy Mosaic on telephony piqued your curiosity, and you'd like to find out a whole lot more about this burgeoning technology and its market, you might want to attend the Computer Telephony Expo, sponsored by *Computer Telephony* magazine, at the Dallas Convention Center on March 8, 9, and 10. For more information about the show, call (toll free) 800-LIBRARY (542-7279) or 212-691-8215.

---

## Apple and Micro- soft to Provide Cross-Platform Messaging and Collaboration Solutions

Microsoft Corporation and the AppleSoft Division of Apple Computer, Inc., recently signed a comprehensive agreement to ensure compatibility of their messaging and directory services. Both companies will also provide their customers with the necessary

software for sharing information across the Apple Macintosh and Microsoft Windows platforms. The agreement focuses on four key aspects:

- access from each platform to the other's messaging and directory servers
- compatibility between the two vendors' messaging and directory servers
- support by Microsoft of Apple Open Collaboration Environment (AOCE) capabilities in its Macintosh productivity applications
- support on both platforms for Common Mail Calls (CMC) application programming interfaces (APIs), resulting in easier cross-platform development for application developers and more choices for customers

Both companies believe that providing compatible messaging and collaboration services in the operating systems will result in customers gaining the benefits of an integrated user interface. In addition, developers will gain the benefits of a common set of APIs that ensures compatibility between client and server products.

Apple and Microsoft have agreed to develop a suite of Messaging Application Program Interface (MAPI) service providers and AOCE-based gateways to allow customers to build cross-platform client-server solutions. The new gateways and service providers outlined in the agreement will give Windows MAPI-compliant applications access to Apple's PowerShare Collaboration Servers. They will also give Macintosh AOCE-based and PowerTalk-based applications access to current and future Microsoft information management products. Microsoft and Apple also agreed to provide a gateway between their server products and to support the basic

send-capability Common Mail Calls in their products.

### Microsoft to Develop PowerTalk Gateway

Under the agreement, Microsoft will provide a personal gateway that will allow AOCE-based PowerTalk users and applications on the Macintosh platform to access and use the messaging and directory services provided by Microsoft's future information management products. At the System 7 Pro introduction in October 1993, Microsoft committed to providing updated versions of Microsoft Works, Word, PowerPoint, and Excel to support Apple's PowerTalk Mailer capability. These updated applications will be able to use the PowerTalk Mailer to send mail and the standard PowerTalk universal mailbox to receive it.

### CMC Support

Under the agreement, Apple intends to support a subset of CMC, a set of common cross-platform APIs from the X.400 Application Program Interface Association (XAPIA). Microsoft has already delivered developer libraries for CMC access to MAPI and plans to include these APIs with MAPI in a future version of Windows. By having CMC support available on both platforms, corporate and professional developers will be able to use a common API to create mail-enabled applications on both platforms. Apple intends to provide support for specific aspects of CMC within PowerTalk.

### Apple to Develop MAPI Service Provider

The agreement also includes Apple's plan to provide a MAPI service provider that will allow applications and users of Windows and MAPI to access Apple's PowerShare Collaboration Servers. Apple also intends to

ensure the availability of a PowerTalk gateway to Microsoft Mail for AppleTalk. In addition, Apple plans to provide migration tools for those current customers of Microsoft Mail Server for AppleTalk who wish to move to Apple's PowerShare Collaboration Servers.

### Joint Development of New Server-to-Server Gateway

Under the agreement, Apple and Microsoft intend to jointly develop a server-based gateway that will allow Apple's PowerShare Collaboration Servers and Microsoft's future information management products to access each other's messaging and directory services.

Apple and Microsoft will announce specific products, features, capabilities and schedules throughout 1994, and *Apple Directions* will keep you posted on future announcements. ♣

# Technology

## Inside This Section

<b>PowerShare—The Workflow Genie</b>	<b>13</b>
<b>Secrets of Authentication Revealed!</b>	<b>14</b>
<b>Human Interface: Fluent Interfaces, Part One</b>	<b>18</b>
<b>The Story Behind the Power, Part Two</b>	<b>20</b>

## Macintosh Computers With PowerPC Available in Third-Party Test Lab

You won't be able to purchase your own Macintosh computers with PowerPC until they're introduced, but until then you can test your PowerPC applications at the Apple Third-Party Compatibility Lab at the Cupertino R&D Campus.

The lab is open by appointment to members of the Partners and Associates programs worldwide for testing commercial applications for the Macintosh computer. Currently, the lab contains more than 50 Macintosh systems, divided among three separate labs, including prerelease versions of the PowerPC processor-based Macintosh computers to be shipped in the first half of 1994.

To make an appointment, call Carol Carrillo at the lab between 1:30 p.m. and 3:30 p.m., Pacific Standard Time, at 408-862-7175; for more information, you may contact the lab at Apple Computer, Inc., M/S 302-1BD, Cupertino, CA 95014; AppleLink: TPC.LAB. ♣

## CD Highlights

### Reference Library Edition, March 1994: *Little Bit Man*

Hello and welcome to the March Reference Library Edition of the Developer CD, featuring 275 MB (count 'em!) of new and revised technical documentation, system software, tools, and utilities.

Along with updates to the ABS Technical Notes, Macintosh Technical Notes, Developer University Course Info, and the Universal Interfaces, this month's featured attraction is the latest release, Beta 3, of QuickDraw GX software. This release provides all the code you need to use QuickDraw GX, plus documentation, development tools, sample

from any folder or volume containing Apple DocViewer files. To use Collector, simply pick up the icon of the folder or volume you wish to create your collection from, and "drop" it on the Collector application icon. Once the program launches, it presents a standard Save dialog box, through which you can specify where to save the resulting file.

#### Developer Notes

Included here, along with our regular archive, are developer notes for a new product: The Macintosh LC 575 computer computer, a cost-reduced Macintosh model similar to the Macintosh Quadra 610/650 computers.

#### HyperCard Player 2.2

The HyperCard Player software enables you to run applications created with HyperCard. HyperCard applications are interactive documents that can incorporate text, graphics, animation, audio, and video.

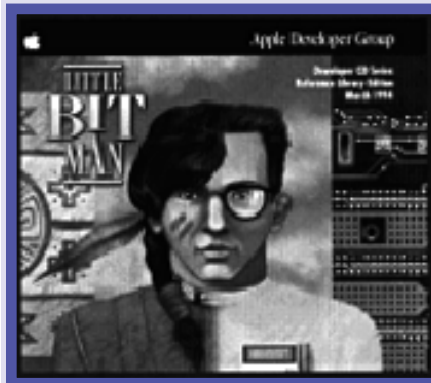
#### International LaserWriter 8.1.1

Included in this folder are localized versions of the LaserWriter 8.1.1 printer driver. Our purpose is to make readily available the localized versions of the LaserWriter driver for companies that have licensed it for use with their products. The LaserWriter 8.1.1 printer driver provides extensive support for Adobe PostScript™ Level 2 PDL.

#### LockVol 1.5

LockVol is a control panel that can be used to test software handling of locked volumes,

*please turn to page 23*



*Little Bit Man*

code, and the *Macintosh Human Interface Guidelines*. We've also gathered the *Inside Macintosh: QuickDraw GX* documents into an Apple DocViewer collection for your browsing and searching pleasure.

Other new and revised packages on this month's CD include the following.

#### Collector

Collector is a beta tool that enables you to create Apple DocViewer "collections" files



# PowerShare—The Workflow Genie

By Kris Newby

Now that Apple Computer, Inc., has let the PowerShare Collaboration Servers product out of the bottle, it's worth considering supporting this group collaboration technology in your software. PowerShare Collaboration Servers (which I'll abbreviate as *PowerShare* just to make it easier to read this article) enables you to create something irresistible to today's "lean-and-mean" companies—software solutions that allow them to do more with fewer employees. "Workflow" solutions—software that automates work processes—have the potential to fill this need, and PowerShare is the behind-the-scenes genie that can make these solutions possible.

To help you understand the potential of PowerShare, this article focuses on one workflow opportunity that this product provides—document approval. Document approval, as unglamorous as it may seem, is a time-consuming, staff-intensive process that can easily be computer automated with PowerShare. We'll explain how you can create workflow solutions like this faster than ever by using the AppleScript scripting language and off-the-shelf applications. And for commercial developers with products that could be used in these types of solutions, we'll talk about how to make them PowerTalk-savvy to take advantage of PowerShare.

## Architecture and Technology

The PowerShare Collaboration Servers product brings the power of collaborative computing to large networks. Based on the Apple Open Collaboration

Environment (AOCE), networked groups using this product with System 7 Pro can take advantage of services like client-server messaging, shared catalogs, authentication, and encrypted communication. PowerShare expands upon the PowerTalk collaboration services built in to System 7 Pro operating software. For a summary of features and benefits made possible by PowerShare, see the chart "How PowerShare Extends System 7 Pro" on page 14.

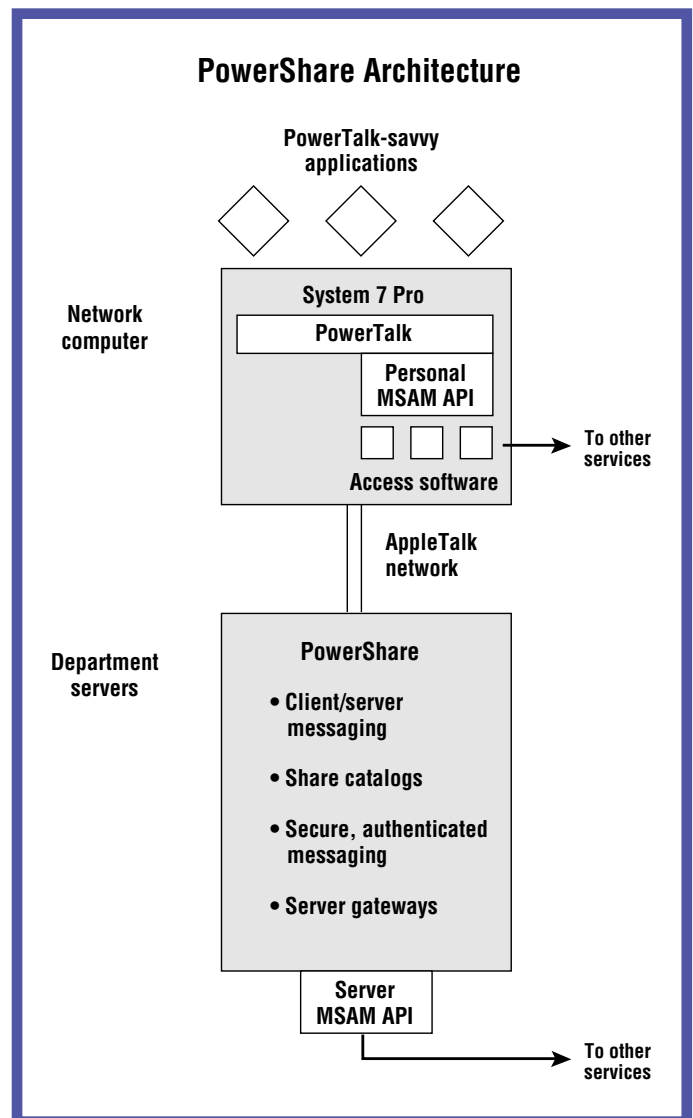
Architecturally speaking, PowerTalk is a different beast than PowerShare: PowerTalk is an integral part of Macintosh system software, while PowerShare is a server-based service that makes some of the PowerTalk features usable across large networks. The diagram "PowerShare Architecture" on this page illustrates this point.

When would a group of System 7 Pro users require the services offered by PowerShare? Well, depending on the complexity of the network, networks beyond 20 users would benefit from this upgrade. Think of PowerTalk as the U.S. Postal Service (peer-to-peer post offices) and PowerShare as Federal Express (distributed routing centers). Because Federal Express only has a few hubs that all mail parcels travel through, they spend less on capital equipment and can more easily track the location of a particular package. Five core components of PowerShare make it useful to large networked organizations:

- *Client-server messaging.* By moving to PowerShare-centralized information services, groups can save money by reducing network administration costs and reducing the system-wide memory footprint dedicated to network services. (This off-loading of

services on the network is particularly advantageous to PowerBook users, who can't afford to overload their limited on-board memory with network software.) For very large networks, it's important that PowerShare is scalable and doesn't have to reside on a single centralized server. This means a PowerShare system can be deployed across multiple, widely distributed servers. Because of this, as organizations using PowerShare grow, they won't have to purchase large, expensive host servers.

- *Shared catalogs.* PowerShare allows groups to create and manage network-wide databases, called *shared catalogs*. These catalogs can contain any type of data for storing information about people, resources, and projects, and can be accessed from anywhere on a network without regard to the location of a particular server. PowerShare's automatic catalog replication feature enables groups to have catalog information automatically distributed across a network of any size,



## How PowerShare Extends System 7 Pro

System 7 Pro/PowerTalk feature	Features added by PowerShare	PowerShare benefits
Peer-to-peer mail and messaging	Client-server mail and messaging	More reliable, efficient mail and messaging across large networks
Catalogs	Shared catalogs	Enables collaborative products
	Distributed synchronized catalogs	Higher performance, minimized network traffic
Key chain passwords	Secure, encrypted mail and messaging	Saves time and money of manually routing sensitive paper documents and thwarts network crackers
	Two-way authentication of communicating parties	Same as above
Personal MSAMs	Server MSAMs	Reduces modems, phone lines; network administration labor; personal and computer gateway memory overhead
	Centralized and richer mail and message queue management	More reliable, efficient communication across large networks

thereby minimizing network traffic and server loads.

- *Two-way authentication.* A PowerTalk user is considered “authenticated” to use various services after correctly entering a “key chain” access code on his or her Macintosh computer. PowerShare extends this protection to AppleTalk networks by providing the network with an authentication service. This service results

in a network where an unauthorized user can't simply steal an access code to gain access to network services or masquerade as an authorized user in a two-way message exchange. PowerShare authentication involves a “key” password exchanged between a client computer, a PowerShare server, and the destination computer in a communication session. (For a detailed

explanation of how PowerShare undertakes two-way authentication, see “Secrets of Authentication Revealed!” beginning on this page.) The user's key chain contains a PowerShare key that is used by PowerTalk's authentication manager to encrypt an initial authentication request to PowerShare. PowerShare uses its knowledge of the initiator and recipient keys to mutually authenticate

communication through a secure protocol.

- *Encrypted messaging.* PowerShare's encryption service ensures that intercepted messages can't be understood by unauthorized users. When an authenticated client wants to communicate securely with another client or server, their keys are combined through a complex mathematical operation and given a time stamp to create a unique session key. This session key is used to encrypt messages for a predetermined period.

- *Server gateways.* Upgrading from personal Message Service Access Modules (MSAMs) to server MSAMs (the software that supplies external messaging and catalog services to networked computers) can save a large organization money: It reduces the number of individual phone lines and modems required, minimizes the RAM required for mail services on individual machines, and cuts down on administration costs for maintaining individual gateways.

### Developer Opportunities

PowerShare makes it easier than ever for developers to create products that help groups of people collaborate. Previously,

## Secrets of Authentication Revealed!

By Rick Andrews

Two people who are separated by a network, communicating with each other through computers, often need to positively identify each other without risk of fraud or impersonation. For example, you may want file sharing to let only certain people view your hard disk, or you may want to be sure that the electronic mail you received really came from the sender named in the letter. This article will explain, without any nasty math or funny symbols like d, e, l, q, y, \$, j, μ, Å, œ, i, •, °, or even √, how the PowerShare authentication service helps people identify themselves to other people with

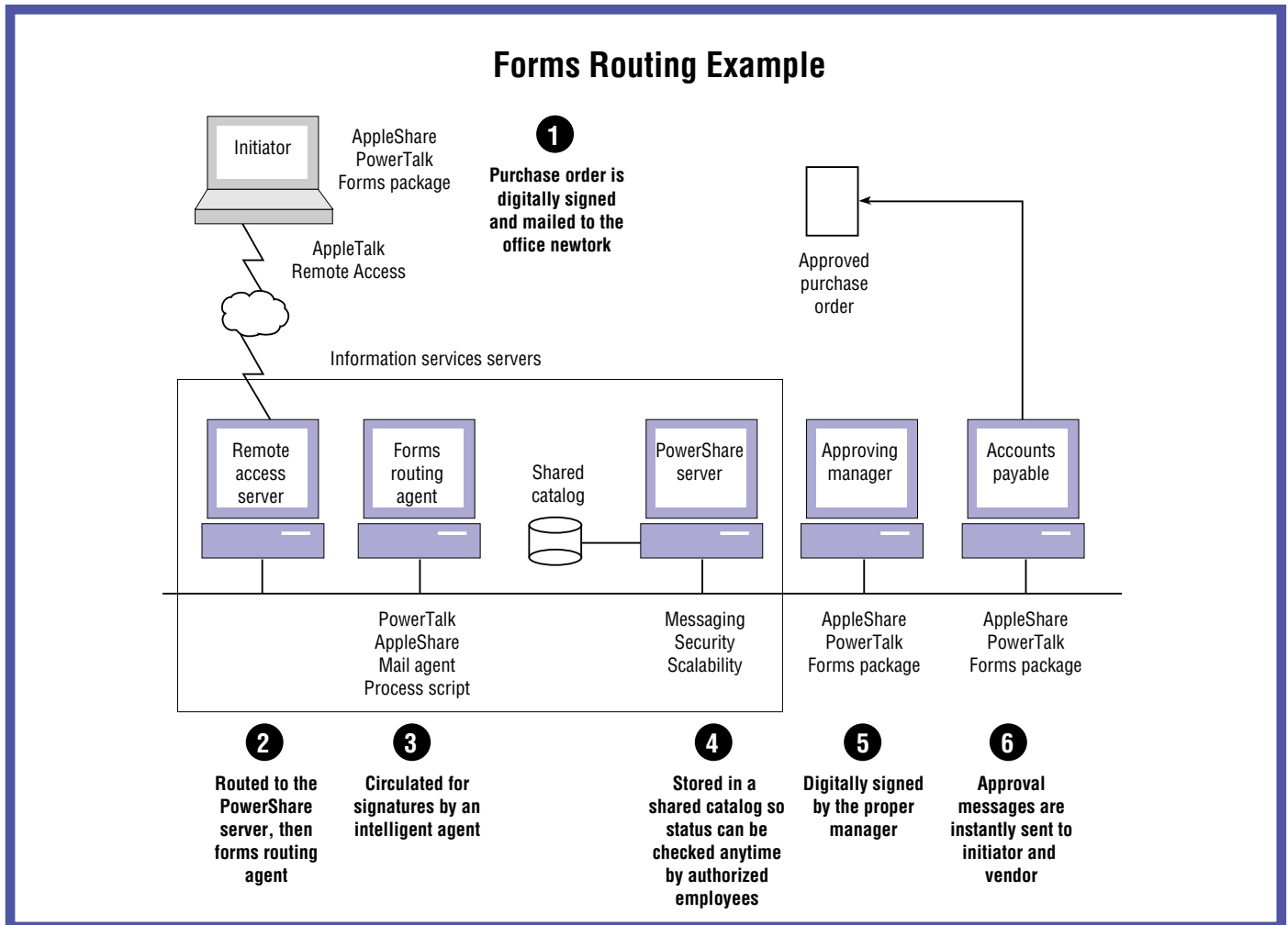
whom they communicate on their network.

In the real (non-computer) world, people take it for granted that when they meet someone they know, they can just look at each other's faces and instantly recognize the other person. In other cases, a person's signature or fingerprint is used as proof of identity. But these kinds of recognition aren't possible on a network. In the stuffy, formal world of networking, recognition of each other's identity is called *mutual authentication*. The authentication service uses a technology called *private-key encryption* to make this possible.

In standard private-key encryption algorithms, a message is transformed by means of

a special algorithm and a secret key into an encrypted message. If the algorithm is strong, then a communicant should not be able to transform the encrypted message back into the original message without knowing the secret key. It's just like locking papers in a strongbox for which only two people have a key. The strongbox can be carried from one person to the other without risk of its contents being revealed to anyone who doesn't have the key.

Once messages are encrypted, they can travel freely around a network. If intruders capture an encrypted message, they cannot decrypt it to discover the original message, even if they have a Cray, or even a Macintosh computer with PowerPC. Nor can intruders create a fake encrypted message, because they don't know the key. If a communicant decrypts



developers who wished to create collaborative tools found that it required years of development just to build the underlying

messaging, directory, and security infrastructures—all of this before the user-value components could start taking shape. PowerShare

gives you these basic tools, so you don't have to create them from scratch. Systems managers will be able to implement customized

workflow solutions using off-the-shelf hardware and software as a base for a fraction of the cost of buying large host-based software.

a message and it contains exactly what is expected, then it must have been created and encrypted by someone else who knows the key.

The authentication service's main role is to give out tokens, called credentials, that users or processes can use to convince themselves of the identity of the person or process with which they are dealing. Note that the authentication service doesn't do the authentication for users; it merely gives them the tools to do the authentication themselves.

The basic terminology used is this: An initiator wishes to talk to a recipient, and they each wish to perform mutual authentication. The initiator and recipient have a secret key known only to them and to a trusted third party, the authentication server.

Let's point out some of the fundamental assumptions: (1) Each person or process's key is private. Even though the trusted third party also knows the key, it is still considered private. (2) The authentication server is trusted to reveal the key to no one.

So, for example, if I can somehow prove to you that I know Rick's private key, then I must be Rick. How do I prove to you that I know Rick's key? I could send it to you over the network, but then it would no longer be secret. We need to be more clever than that.

Instead we rely on the strength of cryptography. As mentioned earlier, if the cryptographic algorithms we use are strong enough, then we can assume that no one can forge an encrypted message without knowing the right key. Every message that we can

decrypt and understand is assumed to have been created and encrypted by someone who knew the right key. And if we believe that the person knows a certain key, then we believe that we know who that person is.

The initiator and recipient do not share any key, but if they did, then they could use that key to encrypt everything exchanged between them. You will see that as a by-product of the authentication process, a new key, not previously used—called the *session key*—is given to both sides in a secure manner, so no one else can see it.

I will omit some parts of the process for simplicity and ease of understanding, but essentially here's what happens. The initiator first asks the authentication server for credentials. He gives his name and the name of the

And the new security features that PowerShare brings to the Macintosh platform extends your market to government organizations and others who need to protect sensitive documents.

Forms Routing With PowerShare

As mentioned earlier, one category of software that illustrates the benefits of PowerShare well is forms routing. Most large organizations are buried in paperwork. Forests are destroyed in the name of product invoices, insurance forms, accounting balance sheets, employee information, legal documents, expense reports, document reviews, personnel forms, and so on. Until now, the greatest barrier to fully electronic paperwork has been a reliance on manual signatures and the security (unalterability) they provide. Together, PowerTalk and PowerShare remove this barrier, and, on top of that, enable groups to

- reduce document flow cycle times
• track forms in the process
• improve process reliability

- decrease administrative and paper costs

The following forms-routing example illustrates how a typical purchase order approval cycle could be automated on an AppleTalk network running System 7 Pro and PowerShare. A similar trial at Apple produced dramatic results—purchase-order routing time decreased from three weeks to three days. Review the companion diagram on page 15 along with the following steps to understand how such a system would work.

1. An engineer visiting an out-of-town power supply manufacturer needs to get a purchase order approved quickly so he can keep his project on schedule. At the vendor site he whips out his Macintosh PowerBook, fills out a PowerTalk-savvy purchase-order form, then signs it with his digital signature. He pulls down the Mail menu within this purchase-order application and sends the purchase order to his office network address. Using his PowerBook modem, he connects to his office's local area network through the remote access server.

2. The form is routed to the office's PowerShare server, then to the forms routing agent.

3. The forms routing agent recognizes the form as a purchase order and handles it according to a routing script that defines the process of getting a purchase order approved. It notes the amount of the purchase order, checks the general employee PowerShare catalog to find out who the initiator's manager is and how much he can sign for, and then sends the purchase order to the approving manager's computer.

4. The agent also logs this transaction into the PowerShare server's purchase-order catalog, so finance employees with proper clearance can use this information to forecast expenses.

5. The approving manager opens the purchase order, digitally signs it, and then returns it using the pull-down "Mail" menu.

6. The purchase order arrives at accounts payable, where copies of the signed purchase order are immediately mailed to the initiator (the engineer) and power supply company. The initiator's desktop

agent notifies him through a pager message.

The result? The engineer gets word of the purchase order's approval before he leaves town, the vendor starts building power supplies immediately, and the project stays on schedule.

Building a Workflow System

The three elements of PowerShare that make workflow practical are its

- robust large-network architecture
• shared catalogs that facilitate intelligent routing
• network security

Systems integrators can take advantage of these elements to build workflow systems by following these simple steps:

- Build an AppleScript framework. Write an AppleScript framework that orchestrates your workflow process and ties together all the system components. This "intelligent agent" that you create will scan your forms and make routing decisions based on its programmed criteria.

recipient, and gets back two quantities. The first quantity contains the session key encrypted in the initiator's key. The initiator can obvi-

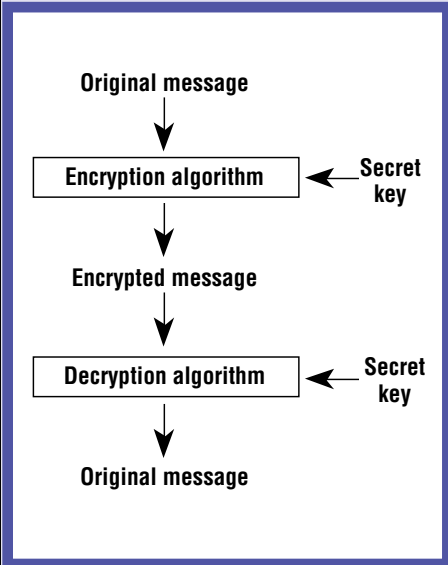
ously decrypt this key and obtain the session key. And of course, if intruders see this message flow over the network, they would not be able to extract the session key.

The second quantity returned is the credentials block, which is encrypted in the recipient's key so essentially no one can see what is in it. The initiator merely sends this block on to the recipient. Once the recipient receives this block, he (and only he) can decrypt it and read what's inside. What he gets is the name of the initiator and his copy of the session key.

If both people are who they said they are, they each have a copy of the same session key. They now exchange a few packets to convince themselves that the other person knows the same session key. This is done as follows: Each person generates a random number, encrypts it with the session key, and sends it to the other person. The other person

decrypts it with his copy of the session key, adds 1 to it, encrypts the new number with the session key, and sends it back. On each side, the originating person decrypts the new number and checks to see if it is the original number plus 1. Because of the strength of the encryption algorithm, the correct result cannot be obtained without knowledge of the correct key.

If this process works on both sides, both people are convinced that the other knows the same session key. The initiator can say, "I'm sure I'm talking to the recipient (and not an impostor), because only the recipient could have decrypted the block I sent and obtained this session key." The recipient can say, "I'm sure I'm talking to the initiator named in my credentials block, because the authentication server gave out the session key encrypted in the initiator's key, and only the real initiator could have decrypted it."





- *Use an off-the-shelf forms front end.* Save time by using a product such as Shana's Informed Foundation as the front end of your system. Shana's new AOCE version features digital signature fields, access to information stored in AOCE catalogs, and Send capabilities using the standard Apple mailer.

- *Tie into a PowerShare shared catalog of employee information.* Create or tie into your company's employee catalog to get information you'll need for workflow—names, addresses, manager names, signing authorities, and so on.

- *Create a process-tracking database program.* Use a simple database like FileMaker to track the progress of each form traveling through your system.

- *Use other communication "glue."* Depending on the configuration of your particular network, you may have to set up other communication gateways, such as Apple Remote Access or an Internet gateway, to make the system work.

If you're a commercial developer, there are two fairly straight-

forward modifications that you'll have to make to your application so that it can be used in these types of solutions:

- *Make it PowerTalk-savvy.*

The two PowerTalk features required for most workflow applications are the Standard Mail Package's mailer, which provides application-level electronic mail support, and the DigiSign digital signature mechanism. (If your users only need to sign an entire document once, you get this feature "free" when you implement the mailer—you have to put in extra work only if users need to sign sections, fields, or cells in a document.) For an excellent tutorial on how to incorporate these two elements, read "Building PowerTalk-Savvy Applications" in the December 1993 issue of *develop*.

- *Make it scriptable in AppleScript.* The better you integrate Apple events into the structure of your program, the more useful your program will be to people creating workflow solutions. The AppleScript bible is *Inside Macintosh: Interapplication Communication*, available at bookstores, through APDA (#T0594LL/A,

\$36.95 in U.S.), or on the September 1993 Developer CD (path—Dev.CD Sep 93:Reference Library:Technical Documentation:Inside Macintosh:IM—Inter-application Comm). Also, Addison-Wesley has recently published *AppleScript Language Guide: English Dialect* and *AppleScript Scripting Additions Guide: English Dialect*. Other AppleScript resources are listed in the December 1993 issue of *Apple Directions* on p. 6.

- *Make it telephony-aware.* At various points in a workflow process, it's common for people to make inquiries of others in the workflow process through the telephone. This is greatly facilitated by applications that support telephony-related Apple events. Apple events are documented on the Developer CD and in both the AppleScript and AOCE developer kits.

### The Upside of "Downsizing"—Solutions Opportunities

As the worldwide trend of corporate downsizing continues, there will be a growing need for software

that helps organizations become more efficient. PowerShare is the network genie that will help you create productivity-enhancing workflow applications for these groups. So, if this genie could grant us a wish, we'd wish that you start incorporating these key technologies, PowerTalk and Apple events, into your software.

### Additional PowerShare Resources

- PowerShare Starter Kit from an Apple authorized reseller
- AOCE Developer's Kit (#R0525LL/A) from APDA
- "How to Get Started With PowerTalk," by Gregg Williams, *Apple Directions*, November 1993, p. 18
- "A Talk With Gursharan Sidhu, AOCE Architect," *Apple Directions*, November 1993, p. 15
- "Building PowerTalk-Savvy Applications," *develop*, December 1993, p. 39 ♣

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

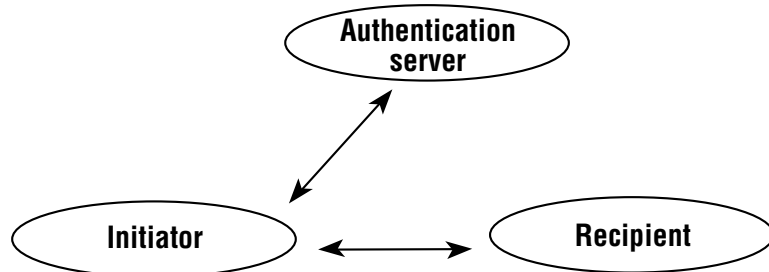
Recall the assumptions stated earlier. Both communicants assume that the only one who knows a certain key is the person or process to which that key was assigned, in addition to the authentication server. Both people trust that the authentication server will not give out their keys to others, or incorrectly fabricate credentials or other data. Furthermore, each person assumes that the other person had to have the session key in order to decrypt, increment, and re-

encrypt the random number. Thus the initiator and recipient have performed mutual authentication. Each is sure of the other's identity, and they have a shared key that can be used to encrypt any information flowing between them.

Once authentication has been performed, the two communicants have a basis for doing authorization—that is, checking to see if they wish to allow the other person to do whatever they request.

As an example, take the current AppleShare server product (please!). When you log in to it, the server authenticates you. When you try to open one of the server's folders, the server checks to see if you are authorized to see the folder's contents. The difference between authentication and authorization is subtle but important. The authentication service doesn't do the authorization for you. Rather, it helps you do the authentication, and, armed with the knowledge of the other person's identity, you can consult a list or use some other method to decide if the person is authorized to do what he requests.

*Rick Andrews is a lead engineer on the Apple Open Collaboration Environment team that has produced PowerTalk and PowerShare. A coauthor of Inside AppleTalk (Addison-Wesley, 1990), he is one of the original designers of the AppleTalk network system.*



## Human Interface

# Fluent Interfaces

## Part One: Speaking the Language

By Pete Bickford

In my January column, I railed against what I saw as the typographical abomination of using foot and inch marks ( ' , " ) in the place of the “real” (curved) quotation marks that civilized publishers had been using for centuries. The use of “straight quotation marks” on computers was largely due to the keyboard’s typewriter heritage, where the limited number of keys made such compromises necessary. In my article, I pontificated about such artifacts of the typewriter age deserving to go the way of the dodo and the PC Jr.

Alas, not long after that, my AppleLink mailbox was besieged by readers pointing out the localization problems involved with using quotation marks. The main points seemed to be that: (a) picking the proper international quotation marks is often quite difficult (“ ” is only proper for English; other countries use different characters), and (b) it’s better for the user to see straight quotation marks than the funny symbols that often accompany improper mapping of curved quotation marks on international systems.

So does this mean that your user-friendly pal has led you astray? Did Doc err? Pshaw! Oh, sure these are good points, but they’re dodging the real problem. Whatever the user’s language (even a bizarre one like English), it’s our responsibility to make sure our interfaces speak it fluently. Quite simply, doing any less just won’t cut it anymore.

### The Tyranny of the Typewriter

Of course, doing less used to cut it—especially in non-English markets. Generations of humans have had their expectations lowered by a tyrannical master known as the typewriter, and its scion, the computer keyboard. It began when Sholes designed his QWERTY keyboard to make typing more difficult (to prevent speedy typists from jamming the mechanism) and launched a dynasty of machines that forced people to adjust to the keyboard’s own limitations. In time, entire languages have been altered to make up for the keyboard’s mechanical shortcomings.

Perhaps no country has had a harder time adjusting than Japan. In part, this is because a literate Japanese citizen is expected to be able to communicate using up to four different writing systems, using each at the appropriate time. Kanji is the most commonly used writing system, consisting of several thousand distinct symbols inherited from the Chinese. Kanji symbols map to individual words, requiring a second, phonetic set of symbols known collectively as Hiragana to add inflections like verb endings and other modifiers. A third system, Katakana, is used primarily for foreign words. The fourth character

system is Romaji, which is the good ol’ character set we use in English and other Roman scripts.

To type Japanese documents adequately, you’d need a typewriter capable of typing thousands of different characters. As an alternative, Kana keyboards that could type the phonetic Katakana/Hiragana characters were developed, although the resulting documents were something of a pidgin Japanese, and not really suitable for serious communication. Later, as more computer power became available, “front-end processors” were developed that let you type a word phonetically in Kana or Romaji, and then dropped you into a special dictionary that looked up all the Kanji words that are homonyms of the word you entered. You then picked the proper Kanji and the computer entered it into your document. “And this,” as the Japanese must have asked, “is making it easier?” No wonder the fax machine—not the personal computer—became the premiere means of exchanging written communication in Japan.

Of course, Japanese is hardly the only language to require “workarounds” for writing with a typewriter. The ubiquitous American typewriter brought about the practice of “spelling out” German diacritical marks, such as using *oe* instead of the hard-to-type *ö*. Even then, German fared better than many Norse languages, which saw similar typewriters erase their diacritical marks altogether, simply replacing *ø*’s with *o*’s, and *Å*’s with *A*’s. The age of the keyboard apparently had little use for people whose families were named *Hjartøy* or *Åkkeson*.

Typewriters have forced their share of changes on English as well. Computerphiles may remember the time—not so long ago—when it was common for keyboards to contain only capital letters, leading them to produce no small number of uppercase-only documents. Legend has it that this convention grew out of computers’ use as successors to Morse code transmitting equipment. Morse code only allows for 36 letters and numbers, and a decision had to be made about whether letters should be written in uppercase or lowercase. The “engineering solution” would have been to use lowercase letters, as they are considerably easier to read. Lowercase might have won out, too, if not for one word—a word that the designers could not in good conscience have sent in lowercase: *God*.

### Growing Up

But heck, what’s the big deal anyway? After all, aren’t the Japanese better off with Romaji than with their thousands of hard-to-remember ideographic characters? Wouldn’t French be a lot easier to write

without all those funny swirls and dashes? And honestly, who among us wasn't secretly just a little glad that capitalization was no longer an issue when we turned in school papers written on early computers?

In a way, it's like when your five-year-old child makes you a hand-written card. You'd have to have a heart of stone not to be touched by the badly drawn stick figure and a message like "I LUV U DADY." The message, though misspelled, is certainly clear enough to bring a smile.

In children (and nascent technologies), this sort of thing is really kind of cute. In adults, it's inexcusable. We make smiling exceptions for the mistakes of children and non-native speakers, but there's an incredible pressure to become fluent in order to be taken seriously. And, as if they were children, we've made exceptions in the past for computers—but, in time, we always expected them to grow up. Luckily, that's exactly what they've been doing.

In little more than a decade we've seen the standard in computerized writing go from uppercase-only to mixed case, then go on to include proportional fonts, styled text, special typographic characters, graphics, color, and combinations of different languages and even different script systems—all in a single document. As we reach each new level, it becomes unacceptable to write using the rules of the previous stage. People who WRITE IN ALL CAPS are routinely excoriated on the Internet—a system that itself has only progressed as far as the Shift key. PC users are only recently beginning to feel the heat for writing with monospaced fonts. And Macintosh users, lucky folks that they are, have always had proportional fonts—but now they have people breathing down their necks when they don't use curved quotation marks.

Where's it all going? Frankly, it may soon get to the point where we can write the things with computers that we've been able to write all along without them.

### Great Expectations

The whole issue ties back to one of the fundamental laws of human interface design: Speak the user's language. Those who insist on making the user speak the machine's language must be willing to pay the price in terms of limited acceptance and the loss of sales to the first competitor who does it right.

In terms of localization, Macintosh users have long had a leg up on the competition simply because someone decided it was a good idea to isolate resources (icons, text strings, and so on) from the rest of a Macintosh application's program code. One of the most significant results of this was that it became possible to pass off a program written in English to a translator who could rewrite the text and dialog boxes in French (thus localizing the program for that market)—without having to recompile the program itself. In fact, an application written according to a few simple rules can conceivably be translated into any system that uses Roman characters. Here are some of the more basic rules:

- All text and visual elements should be stored as resources.
- Leave about 30 percent extra space in dialog boxes for the display of messages, many of which become longer when translated into languages such as German.
- Use the international Toolbox routines for sorting and comparing strings.

The next step is to isolate language considerations in our program code. One programmer, for instance, solved the quotation-mark problem by writing a routine that automatically returns the proper pair of curved quotation marks based on the current language system. Other routines, including those of WorldScript and the upcoming QuickDraw GX, help make it easier to generate programs with common core code, but which obey the rules and conventions of any language.

We've been treating the phrase *speaking the user's language* literally up until now, but it also applies in the broader sense. If a program is geared toward accountants or doctors, it should be able to use the appropriate professional jargon. If it's meant for children, it should try to relate to them on their level.

It's no secret that humans work better when their machines adapt to them, instead of the other way around. The very least we can expect, then, is for the interface—the "human" part of the computer—to speak to us in our own language: fluently and without compromise. If we can stop from becoming less than ourselves in order to use computers, we'll have made great strides in achieving computers' true potential: allowing us to do things we couldn't have done otherwise.

*Till next time,  
Doc*

*AppleLink: THE.DOKTOR*

Next time: From crossing continents to crossing platforms—the truth about users and cross-platform applications.

---

*Pete Bickford is a member of the Apple Business systems human interface team.*

# The Story Behind the Power, Part Two

## An Interview With Sheila Brady and Jim Gable

For those of you just joining us, at the end of last month's article Jim Gable and Sheila Brady were telling us about the fast adoption curve Apple Computer, Inc., anticipates among developers for its forthcoming PowerPC processor-based Macintosh systems.

They said that 100 Macintosh applications capable of running in native PowerPC mode are expected to be released within a year of the (imminent) introduction of the first Macintosh computers based on PowerPC technology, and that virtually all of your existing Macintosh applications already run—without alteration—on the new RISC-based systems.

They also talked about the technical side of the project, how Apple's engineers will be delivering new, blazing fast Macintosh computers that promise compatibility with existing systems and Macintosh products as well as dramatic performance enhancements for new applications.

This month, Jim and Sheila continue the story of the PowerPC project at Apple: talking about the market for the new systems, telling how Apple will be targeting current users of DOS/Windows machines with them, and describing the potentially bright new programming future opened up by Macintosh with PowerPC.

Sheila is project leader for the Macintosh system software that runs on Macintosh with PowerPC; Jim is PowerPC product line manager. Read on for a glimpse of the business side of the project.

•••

**Apple Directions (AD):** *Two years ago, Jim told Apple Directions that we'd see the first Macintosh computers with PowerPC*

*in the first half of 1994, and it looks as if that's going to happen. It's a remarkable feat to keep such a large project, with both hardware and software components, on schedule. Can you talk about how that happened?*

**Sheila Brady (SB):** It was actually a dual process, including, on the one hand, reeling in expectations at the beginning for what we could deliver on time and, on the other, a lot of hard work and phenomenal cooperation between the members of the team.

Even given those factors, I find it completely miraculous that we're actually going to deliver a product early this year, considering that the schedule appeared unrealistic to all of us when we first set out. Near the beginning of the project, I went on sabbatical for three months and came back to hear the powers-that-be saying, "The hardware will be able to ship on this date; therefore the software has to be able to ship on this date and that's the end of it and there is no room for discussion. Period. The end."

So, on the software side, we began a huge thrash over the issue of whether we'd be able to translate enough of the operating system into RISC mode to deliver mixed-mode functionality by then or whether we would just deliver the emulator. It seemed impractical to go just with the pure emulator, so people looked at it and figured out what we actually had to translate to offer native operating system performance. Originally, we thought we'd have to take a bare-bones approach and reduce the feature set as much as possible to meet the date. But luckily,

we didn't have to. The system software we're shipping with the first Macintosh with PowerPC computers goes quite a bit beyond that.

**Jim Gable (JG):** To get to where we are today, people had to be very aggressive and move the project faster than we normally do. Initially, there were, as Sheila says, all kinds of arguments about whether it was possible to meet the schedule. Looking back on it, I believe a couple of things happened. One is that all the key members of the team saw how important it was to keep things on track for Apple's credibility, for its competitive situation and because on-time delivery would give us a real advantage in the marketplace. That motivated everybody.

Second, once we got into it a little bit more, all the groups involved developed a tremendous amount of respect for each other. Because of that, people have been prodding each other. There have been tremendous efforts with groups doing things unusually and boosting their normal processes. At every stage of the project, one team or another would be the one under the gun. For a while it would be system software folks, then the people doing ROM, or testing, or hardware. They'd all make their schedules, forcing other groups to make their dates, to stay on schedule and to lift their part of the load.

**SB:** We've focused a tremendous amount of energy on making communications between the groups really work. We've had these hellish Friday meetings every single week for two years now in which all the team leaders talk about what's going on, what they've accomplished, and what

the problems are. We try to make the problems as obvious as we possibly can with a "top issues" list, which is a very public way of saying, "Here are our problems. These are the things we need help on. These are the managers who are not doing their jobs today." It's amazing how effective public ridicule can be!

### PowerPC Sales and Markets

**AD:** *Now that product release is imminent, can you tell us a bit about the market for Macintosh computers with PowerPC? What kind of sales projections is Apple making, and what markets are the computers going to be sold to?*

**JG:** At the May 1993 Worldwide Developers Conference, we announced that our goal for the first year of shipment is 1 million units. A lot of people were skeptical when we said that, but it's still our goal and we can do it.

The markets are the ones you might expect for mid-range and high-end computers—the mainstream business market, from small to medium and large businesses. We expect PowerPC systems to become Apple's standard business platform.

There are a number of reasons for that. First, businesses tend to buy a little further up the line, because they need better performance and the newer capabilities. Also, businesses tend to buy systems that they think will last several years, because they want to bring in a system and have it work for a long time. Also, once they're on Macintosh with PowerPC, business users have all kinds of overhead left over for future software improvements, another important factor in that market.

We think businesses with mixed computing environments,



where there are both Macintosh computers and DOS/Windows machines, will be really interested in PowerPC. Thanks to Insignia's SoftWindows emulation software, Macintosh computers with PowerPC can run Windows software at speeds comparable to Intel 80486 machines. That's going to be perfect for a lot of mixed environments. Customers who understand the benefits of Macintosh computers—such as lower installation and maintenance costs, ease-of-use and ease-of-training—can have all those benefits; if they need a certain vertical application that's only available on DOS or Windows, then they can also run it on Macintosh with PowerPC systems.

Another good market will be computer enthusiasts, people who really enjoy personal computing and like having the newest and greatest. Macintosh with PowerPC is going to be, by any measure, the newest and greatest personal computer for quite some time, so it's going to be a big hit with the computer enthusiast crowd.

It will also be a big hit in our university markets, where Apple already has a very strong market share. We've seen both faculty and students buy substantial numbers of our mid-range Macintosh computers, and we expect that to continue with the PowerPC systems, as well.

And then, I'd say the last area is what I call *professional users*—people whose professions would be impossible without computers. People in this segment are going to just love the new computers! This area includes desktop publishing, multimedia authors, and people doing CAD/CAM, statistical analysis, or financial analysis. For professional users in those and other categories, computing speed directly correlates to individual productivity over the course of a day. We expect such professional customers to be switching over immediately.

### DOS/Windows Switchers

**AD:** *Which brings up a more specific question: Will Apple focus first on selling the new computers to the existing Macintosh installed base or on converting DOS/Windows customers?*

**JG:** We'll be doing both. The initial focus will be on the installed base; however, one of the main reasons we're coming out with Macintosh with PowerPC is to grow the installed base and attract DOS/Windows "switchers." You'll see us advertising outside traditional Macintosh markets and outside traditional Macintosh channels. It's a fairly simple story: Macintosh with PowerPC will offer superior price/performance. With SoftWindows, Macintosh with PowerPC systems can run DOS and Windows applications more easily than, and at least as fast, as many existing Intel 60486-based systems. With those two parts of the story, we offer a compelling migration path to DOS/Windows customers.

**SB:** A very important part of that group consists of those looking to buy more powerful hardware. We expect that PC customers who have computers with Intel 80286-based systems, or below, will be very interested in the price/performance of Macintosh with PowerPC.

**JG:** Yes, that's one subsegment of the PC world that we'll be going after. Another is the group of DOS users who have yet to upgrade to Windows. We expect they'll be very interested in Macintosh with PowerPC because, for a very reasonable price, they can get great performance, the Macintosh benefits, and the ability to still run their DOS programs, all on the same computer.

We're also targeting two other groups within the DOS/Windows market. First is the "work-at-home" market, customers who use PC/Windows systems in the

office but who want a Macintosh computer for working at home because it's easier to use. They don't have system administrators at home, and their kids use Macintosh computers for school. On the other hand, they often bring work home, work that's done originally on DOS/Windows machines, so until now if they've wanted to work at home they've had to buy a PC compatible. Macintosh computers with PowerPC will, obviously, be a great solution for these folks.

The last area of DOS/Windows switchers that looks promising are small businesses. Small businesses don't have system administrators and don't have resources to spend on support and training. So they like the fact that Macintosh computers are easier to learn, easier to use, and cost less to maintain. But they have two concerns: First, vertical applications—for example, a specialized real estate or dental application—may be available only on DOS or Windows. Second, a lot of small businesses are nervous about being frozen into one camp or the other, so there's a temptation to go with the bigger camp, just because it sounds safer. Macintosh computers with PowerPC, since they can run Macintosh, DOS, and Windows applications, address these concerns.

**AD:** *What types of customers have received seed units, and what have their responses been?*

**SB:** I was just going to mention that we've had seed units out to customers for a number of months. Response has been terrific! We've gotten some powerful messages that we've hit our product right on the money. It's compatible. Performance is great. And customers are telling us, "It's there."

**JG:** We've sent it to a mix of customers. A lot of them were cynical about Apple to start out with,

which made their positive responses all that much more valuable.

### Intel's Response

**AD:** *The whole story sounds wonderful, but what do Apple and developers have to watch out for? How is the competition going to respond?*

**JG:** Intel is doing exactly what it did in 1991. In 1991, a lot of DOS/Windows companies said, "We're going to go to RISC with the ACE Consortium." The ACE Consortium blew up, and a big reason why it failed, I believe, is that Intel said, "You don't need to do it. We have a great new CISC chip right around the corner and you won't need RISC." With that kind of pressure and some other factors, the ACE Consortium failed.

Intel's doing some very similar things today. They're already talking to Macintosh customers, saying, "This PowerPC is going to be a very difficult transition, and you should really consider Intel products," which Macintosh customers have a difficult time believing. They're also telling analysts and researchers that PowerPC is not as strong as it sounds.

But everything developers say suggests that PowerPC is, in fact, a stronger technology than Intel's Pentium technology, so Intel is going to the next tack of saying, "Well, we have a great new Pentium chip coming out right behind the current one."

Let's look at their history. The first Pentium chip was late, and when it arrived, it was expensive and complex and it ran hot. Who knows how much better the follow-ons will be? Of course, Intel is, technically, a very strong company; they'll come out with better products than the first Pentium chip over time. But the fact that they're pushing an old architecture will make it very difficult for them to compete with

our brand-new RISC architecture. So as long as they choose to do that, I think Apple is going to have a sustainable advantage.

### Product Plans

**AD:** *We've been hearing that it won't take long for the Macintosh line to go all RISC, perhaps two years. Will RISC pervade the Macintosh computer line?*

**JG:** We picked PowerPC and we did all this work with IBM and Motorola so that, over time, we can be all-RISC, because obviously that's the way to go. But that's going to take years. First off, putting RISC chips in the low-cost systems is going to be a lot of work; to do that we need some of the new PowerPC chips that haven't been released yet. Also, there are some markets that won't go to RISC for some time. So we know we're going to be selling both 680x0 and PowerPC processor-based Macintosh systems for many years.

In 1994 and beyond, we expect to introduce new PowerPC systems as well as new 68040 products. We still sell 68030-based systems today, and people are very happy with them, so we expect that it will be a gradual migration in demand from 68040 to PowerPC. Even after we stop selling 680x0-based systems, we're going to support them for years beyond that, because we know people will continue to use them for a long time.

**AD:** *Yes, we suppose Apple has to keep in mind its current installed base of, what, about 13 million computers.*

**SB:** Yeah, really. The installed base remains the greatest market opportunity.

**JG:** That's exactly the point; we'll support both systems. One of the benefits of the new computer is that, from the user's point of view,

it runs just like every other Macintosh. That will make it much, much simpler to support both PowerPC and 680x0 product lines.

**AD:** *How much can you say about Apple's specific plans for new products in 1994?*

**JG:** Maybe a comparison to our 1993 product mix will help. If you look at our product line last year, in the spring, basically all Macintosh systems from the mid-range on up used 68040 chips. By the end of this calendar year, from the mid-range on up we'll be offering Macintosh with PowerPC.

PowerPC processor-based systems won't exactly be replacing 68040 computers. Today the 68040-based Macintosh Quadra 605 is selling for under \$1,000, and we won't be able to hit a similar price point with Macintosh computers with PowerPC right away—although they'll be very aggressively priced right from the start. What's happening is this: Our Macintosh product line is currently a mix of 68040 and 68030 systems; in 1994, it will be PowerPC and 68040. Then next year, in 1995, we'll begin hitting lower price points and you'll see the first PowerPC processor-based PowerBook computers.

### Key Points for Developers

**AD:** *If you were speaking to a room full of developers right now about Macintosh with PowerPC, what would be the most important points you'd want them to take away?*

**JG:** First, that the project is meeting its goals, in terms of the schedule, compatibility, and performance: I can say definitely that the first PowerPC processor-based Macintosh systems will ship in the first half of the year, and even sooner than most people anticipated.

Regarding compatibility, we set very high goals and we are

completely on track. The emulator is incredibly rock-solid. Compatibility is also very high with the rest of the system, and it extends to hardware; peripherals, like scanners, hard-disk drives, monitors, and NuBus cards plug right in and work right away.

About performance, developers who've begun taking applications native can probably tell the story best. People who are porting major applications—stuff that people buy and use today—are seeing two to four times faster performance on Macintosh with PowerPC seed units than on Macintosh Quadra computers. Even the low-end PowerPC beats the high-end Quadra in running most native software.

Another important message for developers, and customers, too, is that the new systems are Macintosh computers, very good and blazing fast Macintosh computers, but they're still Macintosh. Their features will be similar to the high-end Quadra computers that we shipped in 1993; they'll support the AV technologies that we introduced last year. So if developers have projects that take advantage of speech recognition, phone integration, or video integration, that's all available on PowerPC.

**SB:** It's interesting to note that the way we've implemented some of the AV technologies is directly on the floating-point unit of the PowerPC 601 chip, negating the need for the DSP chip entirely. And we've found the performance is terrific, even on the low-end 601.

**AD:** *How will developers be behind if they don't adopt PowerPC today?*

**JG:** Some won't be behind at all for the time being. For example, if you develop for the K-12 education market, there's no reason to jump all over PowerPC right now, because for now sales of PowerPC

processor-based Macintosh computers in K-12 will at first be slow. Same thing if you do primarily PowerBook development, since PowerBook computers won't use PowerPC microprocessors until sometime in 1995, or if you do terminal emulators. Terminal emulators don't get faster on PowerPC, although you could do compression faster.

However, what I would say is this: Even if you think your area won't immediately be affected by PowerPC, say if you're a K-12 developer or if you focus on mobile products, there may be aspects of your product line that could be updated right away. Somebody who focused on mobile products for PowerBook users might have a desktop-server part of their product line that runs on desktop Macintosh computers that could benefit from PowerPC performance.

More important, developers in those categories should be thinking about next year, because, as I've said, PowerPC will soon reach lower-end, lower-priced machines as well as the PowerBook line.

**SB:** Now is the time for them to start exploring the user interface, for example, and how they can improve their customer's computing experience using higher performance. That's a very time-intensive sort of effort that any developer really can devote the time to now, to start thinking through what they want to do and what they can actually do.

**JG:** Right, because now, for example, with the tremendous floating-point unit, you can do more realistic things in 3-D than you ever could before. That's probably going to revolutionize more than a couple of games and educational packages.

If I were doing K-12 development, I'd be thinking, "Well, QuickTime is now running much faster and my software will run

much faster, so maybe I want to start bringing even more sound and animation directly into the product." The new performance opens up a great deal of potential to do things that have never been done before.

### Setting a New Baseline

**AD:** *Which brings us to our final question. We've talked a lot about taking current applications native, so that existing products can take advantage of the increased performance. What's the potential for brand-new applications, written explicitly to take advantage of the speed and features of Macintosh with PowerPC?*

**SB:** That's one of the most interesting things for me. Today, a lot of developers are thinking, "I've got my C code. Let's slam it over to PowerPC. Let's get it running and we'll see this performance improvement." I think developers' brains will soon recognize just how much extra potential the new architecture gives them. They'll be thinking things like, "Well, wait a minute. What about those user-interface modifications that I just discarded because they were too complex? Now maybe there's enough processing power to really implement them."

As developers start to recognize the potential of Macintosh with PowerPC and apply it to the user experience, I think we're going to see different kinds of applications. And that, I think, is what's really interesting for the future.

**JG:** Absolutely, and that's another reason why it's a strong foundation. Today, when you design a Macintosh application, you have to think about what's it going to be like if the user puts it on a Macintosh Classic® computer or even a Macintosh Plus, right? A lot of developers today have to take that into consideration when

they're designing applications.

Now, if you design an application just for Macintosh computers with PowerPC, you can say, "Well, my baseline is a 60 MHz RISC chip. Color machine. Lots of memory. Can do voice recognition and the other AV features." A tremendous contrast. And that will be the new baseline for years to come. I think that's very exciting.

**SB:** I think that's a critical point for the developers. Now, with PowerPC, there's room to grow; there's breathing space. I think that there's a mindset among developers that says, "I have to write for the lowest common denominator."

For people designing products just for PowerPC, the new lowest common denominator is not very

low! We're expecting the adoption of Macintosh with PowerPC to be fast enough that some people will take the initiative to break with the old mindset and say, "This is my new baseline." These folks will go out and write brand-new applications that do incredible things with the new technology. ♣

## CD Highlights

*continued from page 12*

local or remote. LockVol displays a pop-up menu containing all volumes in use by your Macintosh and allows you to change the status of each volume from unlocked to having a software or hardware lock. Changes made in the control panel are immediate and remain intact until you change them back or reconnect to the affected volume. The status of the startup volume is restored after you restart the system. All other volumes statuses are lost after a restart.

This version includes the following improvements:

- LockVol now ignores invalid volumes that appear in the VCB queue. For instance, when LockVol is used with System 7 Pro (for example, PowerTalk extensions), the Mail Enclosures volume is no longer included in the volume pop-up menu.
- A little-known limitation has been removed: LockVol now supports an unlimited number of volumes (depending on memory and system software limitations). It previously only supported 30 volumes.
- Internal changes were made to speed up the building of the volume list and processing of selections from the volume pop-up menu.
- Code size has become smaller yet.

### Lurkers

Lurkers searches a directory for files that can be modified, including files not in a project, files checked out for modification, and modify-read-only files. It's much, much faster than the old FindLurkers script; Lurkers takes one minute to search the entire Finder source tree, while the old script takes forever.

### Network Launch Fix Extension

This extension fixes a bug in Macintosh system software versions 7.0.1, 7.1, and 7.1.1, which caused 68040-based CPUs to crash when programs were launched over a network.

The enclosed extension causes the 68040 instruction cache to be flushed on synchronous

read instructions. This will prevent a problem that occurs when locked code segments are launched over a network or over a foreign file system (such as ISO 9660).

### OpenDoc

This folder contains a technical summary of the OpenDoc APIs and information about how to receive prerelease OpenDoc software. The 60-page *OpenDoc Technical Summary* is the starting place for learning more about the OpenDoc architecture.

### ShowDialogBoxes 2.0

ShowDialogBoxes is a tool for displaying dialog and alert boxes as they appear during run time.

### Worldwide Market Opportunities

The Worldwide Market Opportunities data sheets provide information on the personal computing market, distribution, localization, major magazines and publications, user groups, country information, and more. These data sheets are produced by Apple's Developer Support Center in the United States.

### Coming Next Month

For April, we hope to be able to bring you *Inside Macintosh: Networking*, Macintosh on PowerPC documentation, a boot-everything System Software folder, and the usual bunch of new skanky hacks 'n' stuff. See you there!

*Alex Dosber*  
Developer CD Leader

*Careful readers might notice the word acting missing from Alex Dosber's title this month and last; having accepted a full-time job from Apple, he's now the official Developer CD leader after acting in that capacity for six months. Please join me in welcoming him. You can contact Alex on AppleLink at ALYX.*

# Business & Marketing

## Market Research Monthly

### Configuration Data, Part Three: RAM in the Global Installed Base

With a look at random access memory (RAM) in the Macintosh installed base, we conclude our three-part report based on proprietary data from the Apple Computer, Inc., study of Macintosh customers across the globe conducted in spring 1993. Two months ago (*Apple Directions* January 1994) we covered the use of color monitors in the installed base, while last month (February 1994) we looked at monitor size.

An important factor in designing new (or updating existing) Macintosh products is how much memory your target customers' systems have available to run your software. That factor can also pose a challenge to you because of the increasing size of Macintosh system software and the difficulty of executing advanced features with limited amounts of memory.

#### RAM on the Increase

The good news for you is that the amount of memory in the installed base is increasing: According to Apple's proprietary 1993 customer research data, the average small business user's system is configured with 5 MB of RAM, while large and medium business and government users employ an average of 8 MB of RAM. (See "Average RAM in the Macintosh Installed Base, Spring 1993.")

In addition, Apple's customer data shows that high-end Macintosh systems are set up with larger amounts of RAM; notably, Macintosh Quadra 900/950 computers are installed with an average of 24 MB of RAM. See "Former Macintosh Product Line: Average RAM Configuration" for the average amounts of RAM installed on the models covered in Apple's spring 1993 customer study.

Another sign that memory is on the increase: Macintosh computers currently ship

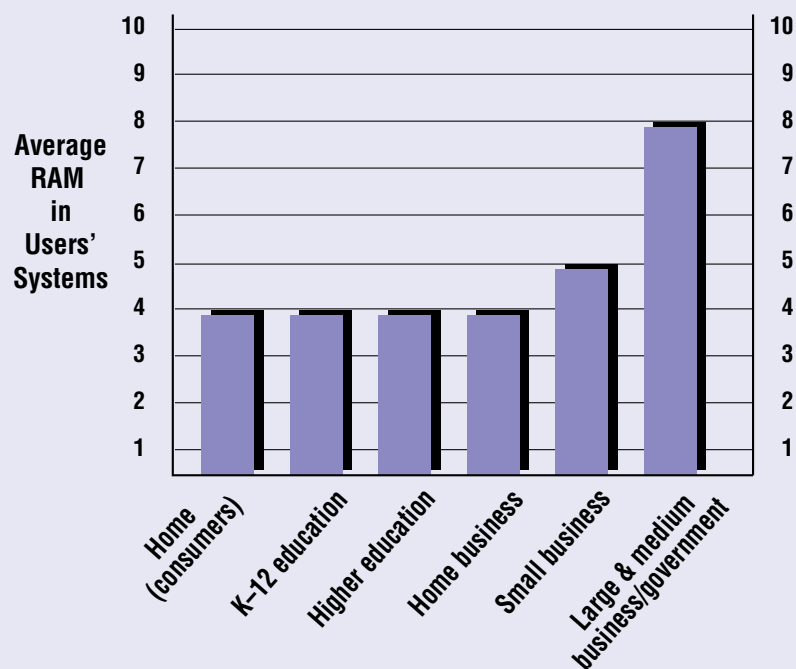
standard with more memory than ever before. "Today's Macintosh Product Line: Base RAM Configuration" shows the lowest amounts of memory available with today's Macintosh computers. When you compare this chart with the information in "Former Macintosh Product Line," keep in mind that "Today's Macintosh Product Line" shows the least amount of memory available; in some cases, it's as much

## Inside This Section

Marketing Feature 1: Your Ten Biggest Competitors, Part Two	26
Marketing Feature 2: Winning the Turf War	29
APDA Information	32

as the average amount of RAM in use by the models covered in the spring 1993 study. Given that today's computers are shipping standard with larger amounts of RAM, the average amounts of RAM in the installed base can be expected to increase over time as users upgrade to higher amounts of memory.

**Average RAM in the  
Macintosh Installed Base, Spring 1993**



Source: Apple Computer, spring 1993



You also need to know, though, that in four of the six major Macintosh market segments throughout the world, the average system is set up with 4 MB of RAM, according to Apple's customer data. If you target customers in the home business, consumer, K-12, or higher education market segments, you can expect them to have a small amount of RAM to run your applications, especially since System 7 currently uses a minimum of 2 MB.

To sum up—yes, buying habits in the installed base are, by themselves, making life easier by increasing the amount of memory available to run your software. But given that RAM will always be a limitation, whether a system is installed with 4 MB or 40 MB, you'll be wise to make sure your product is useful even in circumstances where memory is insufficient to run the entire application. See the text box "Handling Low-Memory Situations—the Friendly Way" on this page for some ideas about how to do just that. ♣

## Handling Low-Memory Situations—the Friendly Way

There are bad ways and good ways to handle situations where your customers' computers have limited amounts of RAM, exemplified by two popular products currently selling to home users. The first is an award-winning "edutainment" product sold by a very successful developer, which you wouldn't know if you tried to start up the software with 2 MB or less of available RAM. Even though the product requires 2 MB of RAM, meaning that, theoretically, it should run on a 4 MB system running System 7, under such conditions it just won't work, telling the user through a dialog box that there is insufficient memory (ack!).

Another home product, an ingenious puzzle game, handles this situation differently. Although it, too, requires 2 MB of RAM, when you start the game up under the conditions just described, it works just fine; parts of the application that aren't absolutely vital to the user's experience (color and sound) are disabled when there isn't enough memory to run the whole thing, and it slows down under extreme conditions. But it works.

Which experience would you like your customers to have? If you want them to have the latter, your software can use the Gestalt routine to check for the total amount of memory in the computer, and it can employ Memory Manager routines to find out how much RAM is available when the application is launched. Based on that information, your application can disable certain features that aren't crucial to the user's ability to work or play, letting your customers take advantage of the product they just bought, even in low-memory situations.

### Former Macintosh Product Line: Average RAM Configuration

Product Line	Model	Average RAM
<b>Low-end</b>	Macintosh Classic	2 MB
	Macintosh Classic II, LC, LC II	4 MB
	Macintosh Performa	4 MB
<b>Mid-range</b>	Macintosh IIsi	5 MB
	Macintosh IIfx, IIfx	8 MB
<b>High-end</b>	Macintosh IIfx	8 MB
	Macintosh Quadra 700	20 MB
	Macintosh Quadra 900/950	24 MB
<b>Low-end PowerBook</b>	PowerBook 140/145	4 MB
<b>Low-end PowerBook</b>	PowerBook 160/170/180	6 MB
	PowerBook Duo 210/230	5 MB

Note: This is a partial list.

This chart shows the average amounts of RAM installed in Macintosh models covered in the spring 1993 Apple customer study. In most cases, the data indicates that customers upgraded their systems with additional memory from the standard, out-of-the-box configurations. We include each product's relative position (low-end, mid-range, and so on) to help you compare the data in this chart with that shown in the adjoining chart, "Today's Macintosh Product Line: Base RAM Configuration."

### Today's Macintosh Product Line: Base RAM Configuration

Product Line	Model	Average RAM
<b>Low-end</b>	Macintosh Color Classic, LC III	4 MB
	Macintosh Performa	4 MB/5 MB
	Macintosh Quadra 605	4 MB/8 MB
<b>Mid-range</b>	Macintosh Quadra 610	8 MB
	Macintosh Quadra 660Av	8 MB
<b>High-end</b>	Macintosh Quadra 800	8 MB
	Macintosh Quadra 950	8 MB/16MB
	Macintosh Quadra 840Av	8 MB/16MB
<b>Low-end PowerBook</b>	PowerBook 145B/165/180c	4 MB
<b>High-end PowerBook</b>	PowerBook Duo 230/250/270c	4MB/12 MB

Note: This is a partial list.

This chart shows the standard amount of RAM installed in most of the current (February 1994) Macintosh product line. The data, culled from the latest list of Macintosh products, shows that in general today's computers ship—without modification—with as much memory as the average amounts of memory installed in comparable models shown in "Former Macintosh Product Line: Average RAM Configuration."

## Marketing Feature 1

# Your Ten Biggest Competitors, Part Two

## To Beat Them, You Have to Know Them

By Regis McKenna

*Editor's note: This article concludes last month's discussion about who your competitors really are and how to beat them. (For a complete list of these competitors, see the text box "Who Your Competitors Really Are" on this page.) Part One from last month covered the first five on the list, so this month we open with competitor 6.*

### Competitor 6: The Bigness Mentality

Edward Schumacher, the economist, was certainly right when he coined the phrase "small is beautiful." Just consider the following statistics:

- More than half of the innovations in the United States in the last twenty years have come from companies with fewer than 200 employees.
- A study by Massachusetts Institute of Technology professor David Birch showed that

companies with fewer than twenty employees created 60 percent of all new jobs, and companies with fewer than 500 employees created 86 percent of all new jobs.

- Of the 9 million jobs created between 1966 and 1977, six million were created by small business, three million by government and zero by Fortune 1000 companies.
- Small companies are more efficient with R&D. A study by the National Science Foundation showed that small companies (those with fewer than 1,000 employees) produced four times as many innovations per R&D dollar as medium-sized companies (those with 1,000 to 10,000 employees) and 24 times as many innovations as large companies (those with more than 10,000 employees).

Indeed, study after study show that small companies are more innovative and productive than larger companies. Unfortunately, as small companies grow and

become large companies, most of them run into the same problems as other big companies. They become less creative and less dynamic. They begin to suffer from what I call bigness mentality.

A major element of bigness mentality is an aversion to risk. Small companies can't afford to take the safe path. They could not compete with established companies on that basis. They must come up with new ideas, experiment with new approaches, try new things. They must innovate or they will not survive.

As companies grow they become more reluctant to take risks. If a company decides to go public, as most do, it is evaluated by the financial community on a quarterly basis. If financial results slip during one quarter, the stock price could plummet and the company could have trouble raising new funds. So public companies must play it safe. They can't afford to take short-term risks, even if they might pay off with long-term benefits. Wall Street thinks short term, not long term.

Corporate bureaucracy also reduces risk-taking and innovation. As small companies grow, they restructure themselves to look and act like big companies. Decisions are made by committees, not by individuals. As a result, decisions tend to be compromises, not bold new approaches. People begin to worry more about avoiding mistakes than creating new ideas.

Take advertising decisions. Advertisements developed by small companies tend to be much better than those developed by big companies. They are more creative, more aggressive, more

interesting, more attention grabbing. Why? Big companies usually have large advertising departments that make decisions by committee. All decisions must be supported by extensive research. People are not willing to stick their necks out.

As a growing company adds new committees and new levels of bureaucracy to its ranks, it is slower to notice new opportunities in the market and slower to respond to changes in the market. Its corporate reaction time shoots up. Earlier [in the book from which this is reprinted] I discussed Intel's quick reaction to the challenge from Motorola's 68000 microprocessor. Within seven days Intel designed a new strategy, presented the plan to 150 managers from around the world, and began to put the plan into action. At most large companies it probably would have taken seven days just to arrange the initial meetings.

How can companies avoid succumbing to the bigness mentality? One way is to maintain small, entrepreneurial project groups within the company. IBM took this approach when developing its personal computer. The company created an independent group in Boca Raton, Florida, and gave the group an unusual degree of freedom.

In doing so, IBM acted in an un-IBM way. Although IBM was still a big company, it was thinking like a small one. It broke its own rules, and took some risks. The risks certainly paid off.

Companies also should avoid compartmentalization in the corporate organization. Many growing companies break various

## Who Your Competitors Really Are

1. Change
2. Resistance to change
3. Educated customers
4. The customer's mind
5. The commodity mentality
6. The bigness mentality
7. Broken chains
8. The product concept
9. Things that go bump in the night
10. Yourself

functional groups into different divisions, then make it difficult for those divisions to interact. In small companies people in the engineering, marketing, and sales departments talk regularly and exchange ideas. They act more as a team.

"Communication does not work well if the group is very large," says Peter Drucker in his book *The New Realities*. "It requires constant reaffirmation. It requires the ability to interpret. It requires a community." This interaction is vital to developing creativity and innovation, but it is usually missing in large companies.

In his book *The Next American Frontier*, Harvard professor Robert Reich argues that large companies must develop new forms of organization that allow greater interaction among different groups. He writes that "precision manufacturing and custom-tailoring, and technology-driven products have a great deal in common. They all depend on the sophisticated skills of their employees, skills that are often developed within teams. And they all require that traditionally separate business functions (design, engineering, marketing, and sales) be merged into a highly integrated system than can respond quickly to new opportunity. In short, they are premised on flexible systems of production."

Only with this type of flexibility can companies avoid the bigness mentality and maintain their creativity and productivity as they grow.

### Competitor 7: Broken Chains

The business world is full of chains and connections. Processes and products are linked to one another in a great chain that ultimately connects companies and customers. No problem or business decision is isolated or self-contained. Companies get into trouble when they think about one link at a time, focusing on advertising or channels or manufacturing,

without recognizing that all these functions are interrelated. By ignoring the linkages companies end up with a broken chain—and a failed product.

To get more specific, consider the most important chain: the product-customer chain. This chain connects everything in the product-development and marketing processes. It starts with the design and planning of the product. Other links include product development, manufacturing, marketing, sales, distribution, product support, and service. The final link in the chain is the customer. Here, the marketplace is a system that must be designed to meet the needs of a specific group of customers.

All of these links are part of one common process with one common goal: serving the customer. What a company does in one stage of the process can affect what happens in many other stages. Manufacturing affects marketing, and marketing affects sales. If any link in the chain is broken, the primary goal of the chain—serving the customer—goes unfulfilled.

A chain is only as strong as its weakest link, so companies must pay attention to every link. They also must maintain strong connections between the links. Different departments must talk to one another and work with one another. If a company fragments into a bunch of loosely connected fiefdoms, it will lose out to a more coordinated competitor. The problems at Xerox's Palo Alto Research Center, discussed earlier [in the book from which this is reprinted], provide an example. The researchers in Palo Alto were top-notch, but they rarely talked to other groups within Xerox. Thus, the product-customer chain was broken, and products got to market late, if ever.

Another important marketing chain involves what are known as "consumption patterns." These

patterns are, in fact, product chains: they link together the sales of different products.

Computer companies generally don't develop their own applications software, nor do most provide customized product packages or localized service and support. These activities must be done through third parties. Yet

might end up buying nothing. For this reason, retailers usually like to handle full product lines, not just individual hot products.

Another important chain is one linking different markets. Sales of a product in one market influence sale of the same product in other markets. In the personal computer business, for

---

**Sometimes all the products in the consumption-pattern chain are available, but are sold through different distribution channels. That can be just as bad for sales as having a missing link.**

---

the customer looks to the computer suppliers to provide the whole product. The computer company, if it is customer driven, will organize and coordinate all the third-party suppliers in order to assemble the whole product package for the customer.

When infrastructure-pattern chains are broken, trouble is sure to follow. If one link of the chain, say software, is missing, the products and services can't be used. Intel's success in the microprocessor business is largely due to its understanding of consumption patterns. Intel sells not only the microprocessors themselves but also the peripheral chips and development systems needed to put the microprocessors to use. Intel constantly adds new types of peripheral chips and microprocessors, and each new product enhances the sales of the others.

Distribution strategy also plays an important role in product chains. Sometimes all the products in the consumption-pattern chain are available, but are sold through different distribution channels. That can be just as bad for sales as having a missing link. The pieces all exist, but they are not linked together into a strong chain. Customers can't easily buy everything they need, so they

instance, the home and office markets are strongly linked. People who use personal computers at work are more likely to buy them for their homes. The reverse also holds: People with computers at home push for greater use of computers in the office. In many cases parents buy home computers for their children; then they, too, become interested in the machines. Before long, they want to use computers in their businesses. Rather than refer to this as the consumer market, I call it the home-business market.

The education market is another link in this chain. Children who use computers in schools often pressure their parents into buying home computers—usually the same brand used in the school. The university market is also important. Today's college students are tomorrow's decision-makers in the business world. In a few years they could be deciding what types of computers to buy for their businesses. They are likely to buy the same brand of computer they used in college.

Computer companies are now scrambling to take advantage of this linkage, offering computer systems at great discount to schools at all levels. IBM and

Apple have donated thousands of computers to elementary schools, high school, and colleges. Companies do this because they recognize the important effect the linkages between the school, home, and, eventually, the business marketplace can

product concept itself will become a competitor.

Earlier [in the book from which this is reprinted] I mentioned the classic business school example of the decline of the railroads. Had the railroads considered themselves transporta-

access information through inexpensive "dumb" terminals. If people use these networks often, they might buy just a terminal, not a personal computer.

Home televisions also will be competitors. In two-way cable and satellite television systems, subscribers can use their televisions to get various types of information services. They can order airline and theater tickets, check their bank account balances, pay their utility bills, and check stock prices. Televisions will become even more formidable competitors to personal computers as manufacturers begin to build computers right into the sets.

In 1990, Radius, a graphics-systems company in Silicon Valley, introduced a product called Radius TV. The product allows computer users to open a window on their computers and receive direct-broadcast, enhanced color television. Users can simultaneously watch CNN and place buy and sell stock orders with their brokers. Government officials can monitor the activities on the floor of the legislature by watching C-SPAN while they write memos or edit legislation. Products like this will change the definitions of the computer and television industries.

In developing their marketing and positioning strategies, personal computer companies must consider all these new competitors, and try to anticipate other challengers. If they limit their product concept and keep their blinders on, they are sure to be blindsided in the marketplace.

### **Competitor 9: Things That Go Bump in the Night**

No matter how well a company understands its market, it is bound to be taken by surprise sometimes. New technologies, new companies, new applications all can shake up an industry when they emerge with little

or no warning. I call these unanticipated events "things that go bump in the night." Companies don't see them coming. But like the iceberg that sank the *Titanic*, they can do a lot of damage.

There are more things going bump in the night today than ever before. The prime reason is the speed-up in technological innovation. According to one estimate, 99 percent of all technological innovations in the history of mankind have occurred in the past twenty years. Each year more innovations come about than appeared the year before, and each innovation holds the potential for shaking up a company, if not an industry.

The base of scientific knowledge, from which technology evolves, is continuing to grow rapidly. According to some estimates, more than 90 percent of all the scientists that ever lived are alive today. More important, scientific knowledge is being put to use more quickly than ever before. Engineers are constantly shrinking the time it takes to translate scientific advances into new technological products. One researcher, studying twenty major innovations, found that the time lag between scientific discovery and technological product has dropped by 60 percent since 1900.

No company in a technology-based industry is safe from unanticipated bumps in the night. The steel industry, the petroleum industry, even the textile industry all have been jolted by technological change. A decade ago the major pharmaceutical companies thought their industry was fairly mature and stable. Then came the development of recombinant DNA technology, and now dozens of new companies are challenging the products of established industry leaders.

The semiconductor industry has been predicting a major

---

## **According to one estimate, 99 percent of all technological innovations in the history of mankind have occurred in the past twenty years.**

---

have on sales. Rather than think of the home as a consumer market, it is better to think of it as the home-education market. The home-office-school chain is important to the long-term success of most computer suppliers.

### **Competitor 8: The Product Concept**

What do IBM, AT&T, CBS, Dow Jones, and Apple have in common? Five years ago the answer would have been not much. IBM sold big computers and office equipment. AT&T was in the telephone business. CBS was a television network. Dow Jones was a publishing company. And Apple sold personal computers. Today, however, all five companies compete against one another, at least indirectly. All are involved in the information business. All offer equipment and services that enable customers to access information quickly and efficiently. In the future they will compete directly with one another more often.

In this type of environment, companies can't afford to think about their products too narrowly. They must look for opportunities in—and expect competition from—every possible direction. A company with a narrow product concept will move through the market with blinders on, and it is sure to run into trouble. The

tion companies, rather than railroad companies, they might have moved into the airline business. Instead, the railroads stuck to their narrow product concept and watched the new airline companies steal much of their business.

The same situation exists now in many evolving industries. Dow Jones, for instance, does not think of itself as a news service or a newspaper company. It sees itself as an information company. It must provide information in whatever form customers desire, whether written on paper, broadcast to radios, or sent over telephone lines to computer screens.

Similarly, a personal computer company should not view its product simply as a box with a keyboard and a display. If it sees its product that way, the company will have a narrow view of its competition. It will see other personal computer companies as its only real competitors, and it will plan its strategies with a false sense of the market.

In fact, many different products could compete with personal computers. Application-specific devices, such as pocket pagers and stock-quotation devices, are potential competitors. So are computer terminals and touch-tone telephones. Many companies are setting up information networks that allow users to



shakeout for fifteen years now. In the late 1960s many industry experts predicted that the semiconductor industry would soon resemble the auto industry, with only three or four leading manufacturers. They argued that the business was too capital intensive for new companies to join. At the time there were about ninety-five semiconductor companies. But today there are more than 200.

Steve Jobs, founder of Apple, made a similar prediction. He said the personal computer industry was too capital intensive to support new entrants, and he predicted a major shakeout in the industry. Even so, he left Apple and founded another company, NeXT.

Certainly there will be shakeouts, but there is no way the industry will consolidate to a handful of companies, at least not in the near future. There is still plenty of room for technological innovation, and that means plenty of room for new competitors. "Shakeout" and "consolidation" are two words I try to avoid using. They make managements believe

that their industries are becoming less competitive. With few players, the marketing task is easier. Few industries can claim less competition this year than existed five years ago.

There is no way for companies to avoid bumps in the night. But companies can be prepared for them. They can stay humble, expect the unexpected, and react quickly when the unexpected occurs. They can stay close to customers and build solid relationships with all members of the industry infrastructure, including their customers. Change is a market phenomenon. Managements must understand that no company is too big, and no industry is too capital intensive, to be shielded from the havoc caused by technological innovation.

#### **Competitor 10: Yourself**

This competitor is the toughest of all. Machines and products don't compete, people do. People have to spot the market opportunities and take advantage of them.

People have to develop the products and competitive strategies, and allocate resources and develop customer relationships.

There are many ways people end up competing with themselves. When people underestimate their own ideas, just because the ideas have never been tried out before, they are competing with themselves. When, on the other hand, people develop an air of omnipotence and believe they can't fail, they also are competing with themselves. When people are unwilling to listen, when they are unwilling to change, when they are unwilling to experiment, they are competing with themselves.

People must leave themselves open to think creatively. With markets changing so rapidly, managers must be able to analyze new situations and apply creative approaches to them. Old approaches to new problems simply won't work.

Above all, managers must pay attention to their customers. They must listen and respond to them. They must not underestimate

their competition—or overestimate it. And they must continue to experiment. Successful companies are led by people who are never satisfied being second best. Leaders make things happen. They have an attitude, a way of thinking that permeates the company. If managers adopt this pattern of thinking, this frame of mind, they can avoid the biggest problem of all: turning themselves into a competitor. ♣

*Regis McKenna is founder and chairman of Regis McKenna, Inc., an international marketing consulting firm headquartered in Palo Alto, California.*

Reprinted from *Relationship Marketing: Successful Strategies for the Age of the Customer* by Regis McKenna. ©1991 by Regis McKenna. By permission of Addison-Wesley Publishing Company.

## Marketing Feature 2

# Winning the Turf War

## Grabbing Market Share Without Positioning Yourself Against the Leader

*By Geoff Moore, Geoffrey Moore Consulting*

Recently I have been reading in linguistic and literary theory, specifically about a set of ideas called *structuralism*. It makes the point that there are no "positive terms" (absolutes) in language; instead, all language is based on a "system of differences." This is how language is learned, remembered, and utilized. For example, *cow* is understood as being different from *horse* (a cow is about the same

size as a horse, but you don't ride it and it gives milk). To continue this line of thought: A pig is shorter than a cow, smells worse, and yields bacon.

On first inspection this may seem to be an ivory tower philosophy; however, it's of great practical importance in the domain of marketing, because customers understand products in terms of their relationship to other products. That is, we understand a Lexus relative to a Mercedes, a 7-Up relative to a Coke, and a Macintosh computer relative to a PC.

Seen in this light, positioning is a matter of establishing not one's *competitive* set but one's *comparative* set.

This alternative is a much more radical approach than you might recognize at first. As a discipline, product positioning typically has been heavily driven by sales organizations, which by nature are antagonistic toward competition: Sales professionals almost always want to position *them* down, and *us* up. Thus, product positioning, when driven by a sales orientation, becomes focused on your competitor's faults.

But this isn't how customers position products. Customers, as

well as the media and other opinion leaders who help customers evaluate products, want to build useful and reliable maps of the buying universe so that they can make informed, successful buying decisions. Neither group trusts sales-driven positioning, and instead they both seek out customer-oriented positioning information.

### **Differentiate Category Before Brand**

The lesson of structuralism that applies here is that customer-oriented maps are constructed as systems of differences—that is, buying alternatives. And the

question these maps seek to answer first is not “What brand am I supposed to buy?” but rather “What category of product am I supposed to buy?”

For sure, once customers establish what category of product to buy, they will pick one brand over another. That’s when a sales-oriented competition between products takes place—after the customer chooses the product category. In other words, category preference precedes brand preference. Therefore, establishing your product category should take precedence over establishing your superiority over the brands in that category.

Understanding this process creates an opportunity for you to outflank competitors not by positioning yourself as being *better*, but rather as being *categorically different*.

This approach has several advantages. First, once the market buys into the idea that there’s a categorical difference between potentially competitive products, then both you and your competitor are free to develop different

niches without having to slug it out head to head.

It’s important to avoid this kind of boxing match, for several reasons. During your slugfest, you’ll create a lot of customer anxiety about who is telling the truth about whom. You’ll also initiate a price competition that erodes your margins—all of which will add FUD (fear, anxiety, and doubt) to the customer’s buying decision and extra time to the sales cycle.

Second, customers are more likely to listen to a message about *different*, whereas they tend to tune out messages about *better*. All vendors claim to be better than their competitors; that’s the nature of the sales process. So customers have learned to ignore that message. But if each vendor says it is different from others, each conceding a unique focus to other vendors, then everyone—vendors and customers alike—considers themselves better served.

For these reasons, in my view positioning should be primarily an exercise in defining product

categories rather than in one-upping competitors. This means we should look at competitors as allies in defining a set of separate niches that can attract customers to each of us.

Of course, as each of us grows, we may attempt to invade the other’s space—all’s fair in love and marketing—and that will lead to authentic head-to-head competition. But prematurely rushing into head-to-head combat is simply unimaginative and may cost you in the long run.

The consequences of head-to-head competition for any company other than the established market leader are usually catastrophic. Why? Because customers assume that the market leader is the de facto choice in any competition. The thinking goes: If the other product (not the market leader) is so much better, why isn’t everyone buying it instead? Obviously, the answer is: because the alternative product is not better. One hundred thousand Frenchmen can’t be wrong. When you try to imply that they are wrong, *you* are rejected as a malcontent or a wanna-be.

Or, to put this in the market leader’s frame of reference, market leaders get an outrageous set of competitive advantages—premium pricing, shorter sales cycles, better third-party support, and extensive publicity—all of which makes them virtually undefeatable on their own turf.

Therefore, to successfully compete, you must establish your own turf. In this context, positioning is an act of customer communication that defines new turf—your home base—and stakes your claim to it, inviting customers and partners to participate with you in creating and sharing in its rewards.

If this communication is successful, the market will expect you to be the leader in this category—since you defined it. Then,

if anyone else competes in this space, they will have to deal with you as the established market leader.

Once you’ve established a leadership position in your initial target market segment, you can look outward toward expanding your leadership position into adjacent territories.

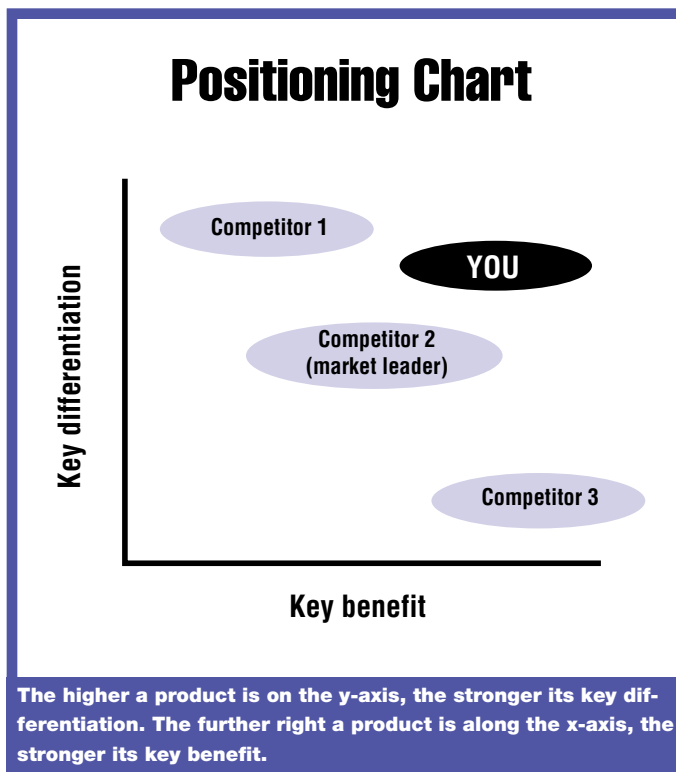
### What Makes Positioning a Challenge

Defining your product category in your target market and developing a leadership position are a real challenge. Again, I turn to language theory for a clue to why this process is so difficult. Languages are inherently conservative. That is, they don’t quickly adopt new words. Instead, a language system tries to define new experiences using the existing system of differences (and more familiar words). Only when an authentically new phenomenon can’t be adequately defined in this way will a language embrace new terminology.

Customers are the same: They resist embracing new categories of products. Customers are prone to pigeonhole you into an existing product category. When you struggle against this, buyers assume it’s the struggle of an “also-ran” trying to escape from a market leader. They may also assume you’re an unnecessary element in the marketing universe—a solution in search of a problem—desperately trying to establish some reason for being.

In short, customers’ first response is to lump you into an existing competitive set—the very thing you are trying to avoid.

Forgive me for putting this in terms of yet another metaphor, but to succeed you must escape the gravitational field created by established categories and the market-leading products that help define them.



**Compare Yourself to Something Outside the Gravitational Category**

Again, it isn't easy to do, but there are many success stories that offer examples of approaches that work.

- *Apple.* When the Macintosh computer first entered the market, it initially was hopelessly positioned against the IBM PC, but Apple was able to penetrate the market by repositioning the Macintosh against, of all things, publishing systems.

- *Intuit.* Quicken was dying as a result of Intuit's efforts to compete with Andrew Tobias's *Managing Your Money*, but the product found new life when Intuit repositioned it against a pen-and-paper checkbook.

- *Sequent.* That company lost out to the technical minicomputer companies in its initial attempt to position itself as a server for workstations. However, Sequent overcame this problem by positioning itself against Digital Equipment Corporation as a superior platform for the Oracle database.

- *Convex.* On a crash course with Digital Equipment Corporation in the engineering market, Convex repositioned itself against Cray, changing its competitive category from "superminicomputer" to "minisupercomputer."

- *Novell.* This company was losing out to the established market leader 3-COM in the LAN systems market, but took the lead when it repositioned itself against the entire industry as a software-only solution.

- *Hewlett-Packard.* In the technical workstation market, H-P lost the UNIX® computer competition to Sun Microsystems, but regained a leadership position in commercial UNIX by refocusing itself against IBM mainframes as an open systems server for commercial applications.

As you can see, these companies have pulled major victories from the jaws of potential disaster.

So the real question is "What line of thinking leads to these new and more productive positions?"

**Explore a New Line of Thought**

To begin with, product positioning must remain simple and focus solely on what the customer base will perceive as being the essentials. The positioning chart on page 30 illustrates these essentials: key differentiation and key benefit.

The important ideas are as follows:

- Customers define categories first in terms of a key benefit. That is, if customers are car shopping, they focus on one key benefit, such as having enough room for kids, luxury, or off-road capability; they use that to identify a subset of the universe of cars to consider. Cars that don't pass this first screening are eliminated from consideration—and thus are not part of that market competition.

- Within the subset of products that passes the key benefit test, there is head-to-head competition. This is decided by virtue of a key differentiator. Going back to the car example, if you picked "luxury" as the key benefit, then the key differentiator could be prestige (Mercedes), styling (Lexus), or price—or "buy American" (Cadillac).

- In the positioning chart, the best position is occupied by "You." Competitor 1 shares your differentiating feature, but is irrelevant to the market competition because it doesn't pass the key benefit test. Another way of saying this is that customers always select products *first* on the basis of key benefits, and only after that on the basis of key differentiation. If you don't pass the key benefits test, you don't get to take the differentiation test. Competitor 1, therefore, is relevant to your position only in a conceptual or academic sense.

- Competitor 2 is typically the market leader. Most market leaders do pretty well when it comes to having key benefits that attract customers, and most of them can claim some participation in all key differentiators. This is what gives them a lot of market space to exploit. On the other hand, they rarely excel in any single dimension. This is what allows for new companies to create and exploit niches around the edges of a market leader's domain.

- Competitor 3 is typically an existing niche leader. It has passed the same key benefit test that you passed, and you are now in an authentic head-to-head competition with this niche leader. The key point here, however, is that you are competing against a niche leader, not the market leader, and therefore you have more "permission" to win the contest. (Fifteen Frenchmen *can* be wrong.)

**Position Against Competitor 3, not Competitor 2**

The key strategy in all this is to position yourself against Competitor 3, not Competitor 2. Here's a recap of the thinking behind this strategy:

- When choosing your turf, don't position yourself directly against the market leader (Competitor 2), because the market is predisposed to

resolve all conflicts in the leader's favor.

- The market leader whose gravitational field you are trying to escape will have a "pretty good" point of differentiation and key benefit. But because you will choose the turf on which to compete (that is, you'll choose the new product category), you'll be better in these specific dimensions. This will allow you to create *niche value*.

- The way to communicate your new value proposition is never to mention the market leader in your positioning messages, but instead focus them against the competitor who best exemplifies your key benefit. This is Competitor 3.

- To make this new niche distinct and autonomous, it should be presented as being a different category from that of the market-leading Competitor 2. In other words, it must be clear that you're not talking about *cars* any more, but about *luxury cars*. As a result, Lexus doesn't have to have the same market niche as Taurus; it just has to be able to compete against Mercedes and Cadillac.

The table "Positioning Examples" illustrates this system using some of the high-tech industry examples cited previously.

The table reads as follows: When entering the market, the Macintosh avoided the gravitational field of IBM PCs by

You	Competitor 1 (lacks key benefit)	Competitor 2 (market leader)	Competitor 3 (niche leader)
Macintosh	Letraset	IBM PC	Linotype
Convex	Floating Point Systems	DEC™ VAX™	Cray
Quicken	Lotus 1-2-3	Managing Your Money	Checkbook

defining a new niche—desktop publishing—which it communicated by asserting itself as a low-cost Linotype. Letraset was another very low-cost solution, but it lacked the key benefit of automation and therefore was only of academic interest in defining a “desktop” category.

Similarly, Convex avoided the DEC space by comparing itself to Cray—again differentiating itself as being the low-cost solution. Here Floating Point Systems was the irrelevant competitor—the product was low cost but wasn’t a number cruncher.

### Summing Up

When positioning your product, remember that customers accept the market-leading solution as the de facto choice. This means that you don’t have “permission” to confront the market leader directly. Chances are, however, that there are many customers who buy the market-leading product

but whose needs aren’t truly being met by it.

The keys to creating a new niche market are to focus on meeting these needs in a far superior fashion and to express this intent to customers by isolating a key benefit that symbolizes this new value.

To do this, try to find an existing niche player who already has demonstrated the value of the benefit to one set of customers, but who, for some reason, will be unable to serve your target segment (because of price, platform, or some other reason). Then position yourself against that competitor—and not against the market leader—as a way of reaching your target customer.

One final point: As your “new” product category catches on, you’ll actually end up stealing customers from the market leader. However, you can’t openly admit that in the marketplace, at least not until you are sufficiently big to win a head-to-head competition.

Instead, you’ll just humbly

profess that you’re sticking to your knitting, and will always be somewhat surprised to learn that your “little niche” is eating away at the market leader’s pie.

Sooner or later, of course, the market leader will come gunning for you, and there may have to be a showdown at the O.K. Corral. But from your point of view there’s no hurry: You’re winning. ♣

*Geoff Moore is the president of Geoffrey Moore Consulting, a firm based in Palo Alto, California, that provides consulting and education services to high-technology companies. He also is the author of Crossing the Chasm: Marketing and Selling Technology Products to Mainstream Customers.*

## More Geoff Moore Available

For a more detailed description of the product positioning process and for a step-by-step process for choosing the right target markets for your products, see Geoff Moore’s chapters in *The High-Tech Marketing Companion: Expert Advice On Marketing to Macintosh and Other PC Users*. This book, by Dee Kiamy and the editors of *Apple Directions*, is published by Addison-Wesley and is available from most major booksellers.

For more information about *The High-Tech Marketing Companion* or to request a copy of its table of contents, send an AppleLink message to Dee Kiamy at KIAMY, or call her at Open Door Communications at 408-266-9699.

# Now Available From Apple

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

### Apple Products

*Macintosh With PowerPC Bundle*  
B1864LL/A  
\$849.00

*Programmer’s Introduction to RISC and PowerPC*  
R0172LL/A  
\$150.00

*Macintosh With PowerPC Starter Kit*  
R0563LL/A  
\$39.95

### Third-Party Products

*CodeWarrior Gold*  
T0937LL/A  
\$399.00

*CodeWarrior Bronze*  
T0938LL/A  
\$199.00

### APDA Ordering Information

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