

Apple Ships Development Tools for Macintosh With PowerPC

We've been telling you for some time to "go native"—that is, to port your applications so that they run in native mode on forthcoming PowerPC processor-based Macintosh computers. With the debut of the Macintosh on RISC Software Developer's Kit (SDK) and a variety of other tools, documents, and courses from Apple Computer, Inc., and Metrowerks at January's Macworld Expo, you can now get going in earnest with PowerPC development.

At Macworld, Apple introduced the following products, which are now available from APDA (further details, including ordering information, are provided later in this article):

- the Macintosh on RISC SDK, a development environment based on the Macintosh Programmer's Workshop (MPW) that runs on 68020, 68030, and 68040 Macintosh computers and generates native code for Macintosh with PowerPC systems
- the Macintosh With PowerPC Starter Kit, a collection of technical documentation that introduces you to the PowerPC microprocessor and System 7 for Macintosh with PowerPC
- the Programmer's Introduction to RISC and PowerPC, a CD-ROM-based self-paced course introducing you to technical issues associated with RISC and PowerPC microprocessor technology
- Metrowerks CodeWarrior Gold, the industry's first development environment for developing applications for both PowerPC processor-based and 680x0 Macintosh computers using the same source code base

In addition, Apple's Developer University (DU) is taking registrations for its four-day PowerPC boot camp, a new course that teaches you how to get your application to run native on the Macintosh with PowerPC platform.

Sixty-One Developers Now Committed to Macintosh With PowerPC

When you take your applications native, you'll join an increasingly large group of developers who have announced plans to deliver new Macintosh with PowerPC versions of their products. At Macworld, 37 additional developers, worldwide, committed to delivering native Macintosh with PowerPC products, bringing to 61 the number of your colleagues who have publicly announced support for the new platform. The 37 developers who announced support for Macintosh with PowerPC at January's Macworld Expo are Abacus Concepts, Inc.; Agfa-Gevaert N.V.; Aladdin Systems, Inc.; Alias Research, Inc.; auto•des•sys, Inc.; B & E Software; Central Point Software; Dayna Communications, Inc.; DeltaPoint, Inc.; Domark Software, Inc.; Electric Image, Inc.; Emergent Behavior; Graftek SA; Gryphon Software Corporation; Imagine That, Inc.; InterCon Systems Corporation; Knowledge Revolution, Inc.; Language Engineering Corp.; Macro Educational Systems; Macromedia; National Instruments; Neon Software, Inc.; Nisus Software, Inc.; Oracle Corporation; ORKIS; Radius, Inc.; RasterOps Corporation; Ray Dream Inc.; Scitex Corporation Ltd.; Strata, Inc.; Storm Technology, Inc.; SuperMac Technology, Inc.; Vibro-Acoustic Sciences, Inc.; VICOM Technology Ltd.; VideoFusion, Inc.; Visual Information Development, Inc. (VIDI); and Virtus Corporation.

Additionally, Apple has announced that it's currently working closely with more than 200 developers around the world on porting existing 680x0 Macintosh applications to the PowerPC architecture.

If you're not yet in that number, *Apple Directions* thinks you'll want to be. Native applications take full advantage of the features and superior speed of PowerPC technology. The first PowerPC 601 processor-based Macintosh computers are expected to deliver two to four times greater performance than the fastest 68040-based Macintosh computer.

Without any modification, existing 680x0 applications will run on the PowerPC platform in emulation mode at speeds comparable to a 68040-based Macintosh, though performance depends on the application. So, your application, as it stands today, is fully compatible with PowerPC processor-based Macintosh computers, and you don't have to do anything to enable your

customers to run your products on the new computers.

But if your competition hasn't yet announced plans to "go native," it soon will, and we don't think you'll want to be left behind. (For more information about the Macintosh with PowerPC program, see "The Story Behind the Power" .)

The rest of this article provides details about Macintosh with PowerPC tools, instructional materials, and courses.

Macintosh on RISC SDK

An MPW-based environment that runs on a 68020, 68030, or 68040 Macintosh computer, the Macintosh on RISC SDK generates native code for Macintosh with PowerPC systems. It includes all of the tools and documentation necessary for creating new applications or porting existing applications to the Macintosh with PowerPC platform. As soon as Macintosh with PowerPC systems become available, developers can test and debug their native Macintosh with PowerPC applications.

The Macintosh on RISC SDK includes

- a C/C++ compiler
- a PowerPC assembler supporting the full PowerPC instruction set
- a two-machine PowerPC debugger, with an easy-to-use interface for setting breakpoints, examining and changing memory and registers, and viewing code
 - universal system header files for both 680x0 and PowerPC processor-based platforms
 - MacApp 3.1, an updated version of Apple's object-oriented application framework, which gives existing MacApp developers a path for porting their applications native on Macintosh with PowerPC
 - Apple Installer 4.0, which installs either 680x0 or PowerPC environments from a common set of files
 - MPW 3.3, which provides a complete development environment
 - additional tools and sample code, including a PowerPC linker, complete build tools and scripts, and sample applications for Macintosh with PowerPC

- complete on-line documentation, including all documentation for the Macintosh on RISC SDK and the PowerPC system software

Most of the tools in the Macintosh on RISC SDK are in prerelease form. If you purchase the kit, you'll automatically receive free updates of these tools, including the final versions when they're ready later this year.

Before Apple ships the first Macintosh computers with PowerPC, you'll be unable to execute the code generated by the Macintosh on RISC SDK. Until then, you can begin the porting process by making your existing code compatible with the new compiler, tools, and system software headers. Once the first Macintosh computers with PowerPC are released, you can test and debug the new code you've prepared. To use the Macintosh on RISC SDK before the release of PowerPC processor-based Macintosh computers, you need to agree to the terms of a confidentiality agreement included with the product.

The Macintosh on RISC SDK, which Apple ships on a CD-ROM, requires a Macintosh computer with at least 20 MB of RAM, a 68020, 68030, or 68040 microprocessor, System 7 or later or A/UX 3.0 or later, a hard drive with at least 20 MB of free space, and a CD-ROM drive.

Macintosh With PowerPC Starter Kit

Apple developed the Macintosh With PowerPC Starter Kit to give you a comprehensive introduction to development on the Macintosh with PowerPC platform. It includes a collection of documents for general information as well as detailed technical documentation about both the PowerPC processor and System 7 for Macintosh with PowerPC.

The kit includes

- *PowerPC 601 RISC Microprocessor User's Manual* from Motorola, Inc., which includes the complete PowerPC 601 instruction set as well as an overview of the PowerPC architecture
- *Inside Macintosh: PowerPC System Software*, which documents the new Macintosh with PowerPC system software, System 7 for Macintosh with PowerPC, including the emulator and the Code Fragment Manager

- *Migrating to Macintosh With PowerPC Checklist*, based on the Programmer's Introduction to RISC and PowerPC self-paced course; the checklist gives you tips for porting an existing Macintosh application to run native on Macintosh with PowerPC

- *PowerPC Technology: An Overview for Apple Third-Party Developers*, which describes the Macintosh with PowerPC platform from a developer perspective, including a high-level overview of the hardware and software architecture and descriptions of the path to PowerPC both for customers and developers

Programmer's Introduction to RISC and PowerPC

The Apple Developer University self-paced CD-ROM-based course Programmer's Introduction to RISC and PowerPC will introduce you to the technical issues associated with RISC and PowerPC microprocessor technology. DU put the course together to prepare you for recompiling existing code for the Macintosh with PowerPC while enhancing speed and portability, as well as for writing new code for Macintosh with PowerPC.

Metrowerks CodeWarrior

CodeWarrior enables programmers to develop applications for both PowerPC processor-based and 680x0 Macintosh platforms using the same source code base. CodeWarrior comes in three versions: Gold, Silver, and Bronze.

The Gold version, currently available in prerelease form, includes development releases of C and C++ for the 680x0 Macintosh and for the Macintosh computer with PowerPC, a development release of Pascal for the 680x0 Macintosh, and C and C++ cross-compilers. (Current purchasers of CodeWarrior Gold will receive the final version free when it's available.) The Silver version supports native PowerPC processor development only, and will be released when Apple ships Macintosh with PowerPC systems. The Bronze version, available now in prerelease form, supports only 680x0 development.

PowerPC Tools Availability and Prices

All the new PowerPC developer products are currently available worldwide from APDA. Prices in the United States are as follows, with prices in other countries varying by location (check with APDA for non-U.S. prices):

- Macintosh on RISC SDK—\$399
- CodeWarrior Gold—\$399
- Macintosh with PowerPC Starter Kit—\$39.95
- Programmer's Introduction to RISC and PowerPC—\$150

APDA is currently offering a special bundle, including the Macintosh on RISC SDK, the Programmer's Introduction to RISC and PowerPC, and Metrowerks CodeWarrior Gold for a U.S. price of \$849 through APDA. You can call APDA from the United States at 800-282-2732, from Canada at 800-637-0029, and from elsewhere at 716-871-6555.

Developer University's PowerPC Boot Camp

DU describes its four-day "boot camp" as a new course for those of you who have existing 680x0 Macintosh code and wish to port it to the new PowerPC processor-based Macintosh systems. Using the MPW-based "fast track" development environment and DU's programming labs, DU instructors will provide you with hands-on experience with the PowerPC development system.

In this course, you'll obtain in-depth information on the PowerPC architecture and instruction set, learn to use the "fast track" development environment, understand key PowerPC internals, learn to port your code to PowerPC, and learn debugging techniques for the PowerPC. DU encourages you to bring your own code to the boot camp.

Three sessions are currently available this spring, February 15–28, April 11–14, and May 23–26, all taking place at Apple's R&D Campus in Cupertino, California. The course fee is \$1600. To register, contact the DU Registrar: phone 405-974-4897; fax 408-974-0544; AppleLink DEVUNIV.

The Newton Takes Off

Notes From the First Newton Platform Development Conference

By Gregg Williams, Apple Directions Staff

Does the Newton platform have a future? Over 700 third-party developers from all over the world think so. Last December 7 and 8, these people (and a huge group of Apple employees) gathered in Santa Clara, California, for the first international Newton Platform Development Conference, where Apple Computer, Inc., and third-party presenters shared market, development, and technical information. Given that I took over 20 pages of detailed notes on the state of the Newton market, the best I can hope to accomplish here is to give you some key bits and pieces on the state of the Newton platform. Fasten your seatbelts—this is going to be the view from 10,000 feet up.

Your first question, of course, is “Is anybody developing for the Newton platform?” Yes. Apple has sold over 2,000 Newton Toolkit (NTK) packages (the software necessary for creating Newton applications)—and with the NTK’s \$795 price tag, I’m reasonably certain that most of the people who have bought it are using it.

“Is the Newton market worth pursuing?” might be your second question. Akira Mitarai is the group general manager of Sharp Information Systems Group, the company that has shipped the first non-Apple Newton device, the Sharp ExpertPad. At the conference’s opening session, he predicted that the market for PDAs (personal digital assistants) will grow to 50 million units by the year 2000—and, of course, Apple plans for Newton-based devices to have the largest single slice of the PDA pie.

Before I get into the detailed content of this column, I thought I'd mention some products that got my attention. They show that Newton devices can be useful in a variety of situations.

- The makers of Tupperware are already using Newton devices to help their salespeople. Their Newton application (meant for in-house use at Tupperware) is more than just a program for taking orders. It also contains other functions that help salespeople do their jobs. One such function is a built-in catalog that includes useful information, including a picture for each product that Tupperware sells. With it, salespeople have enough (dare I say it?) information at their fingertips to sell Tupperware products effectively.

- Qestra, from PRC Realty Systems, gives us just a glimpse of what a Newton device can do when linked to a large database. (We'll see even more when one- and two-way wireless messaging becomes commonplace.) With it, realtors can use a modem to download into their Newton extensive information on local houses for sale from the Multiple Listing Service database. The next day, not only do they have the latest information in a more convenient form—the alternative is a thick stack of computer printouts—but they can also use the Newton to search for the houses that meet their customers' requirements.

Is Qestra a good product? The National Association of Realtors recently voted it the best new product of the year. Will Qestra make money? PRC has 300,000 customers who use its multiple-listing service. If you were a realtor, wouldn't having Qestra more than justify the purchase of a Newton?

- I didn't think much of PocketCall (from Ex Machina, Inc.) when I first heard the idea—a terminal emulator for a Newton device? But I changed my mind when I found out that it contains a terminal-emulation “engine” that can be modified by add-in modules to make a Newton device into the perfect hand-held front end for interacting with a remote database. (Give Ex Machina credit for making its add-in module specification public. That way, others can design modules for their needs and, not coincidentally, sell more copies of the basic PocketCall product.)

- Finally, HealthCare Communications, which sells specialized packages (in the range of \$10,000 to \$40,000) for the health care industry, has begun to use

Newton devices as part of the total solution it supplies. Mark Spector, their director of product development, said that he was amazed when, at a recent trade show, they sold ten systems “on the floor.” He said that the buyers were very excited about the Newton and attributed these sales largely to the value that the Newton adds to their products.

The Invisible Market

Remember how we’ve been told that a Newton device is good for three things that people want to do—collect lots of small bits of information, organize and access them, and “stay in touch”? If the Newton conference convinced me of one thing, it’s that wireless communication will be a much bigger influence than any of us can currently imagine. After all, people want to *stay in touch*.

Gib Hoxie, of the Pacific Group, said it best: “Messaging will be the compelling application for the Newton and other PDAs.” His argument is simple and compelling. Who likes to go to all the bother of hooking a computer (or a Newton device) to a telecommunications service, only to find—after wasting maybe five minutes—that you don’t have any mail? Hoxie calls this “go to” messaging—you have to go to the service to get what you need.

“Come to” messaging occurs when—wherever you are—you’re immediately notified when you have mail. And what better platform for that than a Newton device with a wireless paging card? (And that’s available *today*.)

The only thing better than receiving wireless messages would be the ability to *send* messages, as well. That, too, will soon be a reality. Robert Growney, executive vice president and general manager of the Paging and Wireless Data Group at Motorola, said at the conference that Motorola will deliver two-way wireless modem technology in 1994.

Jay Galvin of Motorola stated his company’s intent to build a Newton device with built-in radio communications, which will make for more efficient power usage and will leave the Newton device’s PCMCIA slot free for other things.

Once two-way messaging becomes possible and affordable, Hoxie sees Newton devices as not only being able to tell you where the best restaurants are (through a wireless information service) but also to letting you make dinner reservations from wherever you are.

Messaging and Mobile Workers

Dr. Alain Briancon, executive director of BellSouth MobileComm, argued that the Newton platform is the best one for developing wireless solutions. First, he said, Newton is an easy platform to develop for. (At the conference, several developers with shipping Newton products gave figures of 3.5 to 5.5 months from initial concept to final product.) Second, the Newton platform makes sense because it already connects easily to Macintosh, DOS, and Microsoft Windows computers. Third, there will be multiple distribution channels for selling your Newton applications. Fourth, Newton supports both one-to-one and one-to-many messaging. Dr. Briancon also had some interesting market figures to share (see “Markets for Wireless Communications” on page 6).

Both one- and two-way wireless messaging will be a boon to mobile workers. And who exactly are they? People who spend a significant amount of time away from their desks, whether in the same building, elsewhere in town, or halfway across the country. In his keynote speech, Apple CEO Michael Spindler noted a change in our orientation to work—from being office workers, where work is provided for us, to being nomadic workers, where “work will be there wherever we are.” Wireless messaging will be one important vehicle (though not the only one) for getting the work to us.

Why a Newton?

OK, so wireless messaging is going to be a big thing for mobile workers. Why not use a notebook computer instead? Don Gummow, senior systems specialist at

Monsanto, offered several good reasons. First, a Newton device is easier to use (because it was *designed* to be used by people on the go), and people are far more likely to take it with them everywhere. Second, managers are more likely to approve the purchase of a Newton device; a notebook computer costs about twice the price of a Newton device—and anyway, why pay for more functionality than you're going to use? Third, the Newton operating system was designed explicitly for supporting PDA applications.

Gummow also added several Newton advantages that relate to communications. He pointed out that the Newton platform has good support for built-in communications; its communications model is generalized and hides implementation details. This means that, further in the future, users can switch to better wireless services, and existing applications will continue to work. Also, the Newton In and Out boxes streamline communication.

Any machine that can work with the existing infrastructure will be more useful than one that cannot. So it's important to note that Newton devices already work well with existing communication devices: pagers, phones and modems, electronic mail and information services, personal computers, a variety of printers, and fax machines. (Did you know that fax outranks e-mail as the preferred way of sending information—39 percent versus 7 percent?) A Newton device's low cost and its ability to connect to fax machines and remote databases make it a good solution in the enterprise (in-house business) sector.

Vertical Markets

If you're in the market to sell to others, vertical solutions offer an alternative to competing in the market of shrink-wrapped products. Jane Curley, Apple's PIE (Personal Interactive Electronics) vertical market manager, listed several vertical markets that PIE feels hold the most promise:

- health care/medical

- pharmaceuticals
- financial
- professional services

The health care/medical market presents several classes of opportunities, she said. These opportunities include electronic claims systems, encounter tracking, medical records tracking, and outcomes analysis.

Curley also gave some advice for those who are looking for vertical market opportunities:

- Understand the way your customers work. If you don't, you probably won't have an adequate, let alone successful, product.
- Broaden your definition of a solution. The Newton is a new kind of device and is meant to be used in new ways. If you continue to think with a personal-computer mentality, you'll miss the compelling new solutions that the Newton can offer.
- Look for alliances with other companies. By working with other companies, you leverage their expertise, get a reality check on what you're doing, and bring your solution to market faster. Vertical-market problems are often too big for you to tackle by yourself.
- (From Mark Spector, of HealthCare Communications) Focus on solving the most profitable problem first, then expand your solution to solve the larger problems as the Newton environment grows.

The Worldwide Newton

Do Newton development opportunities exist outside the United States? You bet! Apple has released separate versions of the Newton MessagePad in the United States, the United Kingdom, and Germany. And Apple has announced plans for Japanese, French, and Swedish Newton MessagePad devices before the end of 1994. I have at least one indication that developers are fairly excited about the Newton platform—over 30 percent of the Newton Platform Development Conference attendees came from outside the United States.

Perhaps the most important thing for you to know about the world market is that, with a modest amount of work, you can sell your English-language Newton software—especially business-oriented products—outside the United States. Canada, Australia, and the United Kingdom are obvious English-language markets, but Hong Kong, Singapore, and Japan also have significant numbers of people who use English-language products. By designing your product for what is called “world English,” you can significantly increase your market.

Here are some steps you should take to make your program ready for the “world English” market:

- Display metric measurements if the market requires it; most countries outside the United States do.
- Use the correct date format (many countries use day/ month/ year), decimal point punctuation (a comma in some countries), and thousands separator punctuation (often a period).
- Be aware of spelling (*colour* instead of *color*), terminology differences (*lorry* instead of *truck*), and slang. Many people from the United Kingdom, for example, are completely puzzled by American sports metaphors.

- Make sure that your software doesn't contain any images or figures of speech that the target audience might find inappropriate or offensive.

For more thorough coverage of localization issues in general, read *Guide to Macintosh Software Localization*, available from Addison-Wesley or through APDA (M1528LL/B; \$24.95 in the United States).

Since the Newton "knows" geographically where it's being used (the user sets this through the Time Zones utility), you can write one piece of software that works in multiple countries. By accessing certain internally stored information (specifically, the "userConfiguration" entry in the soup named "System"), your program can find out what country is current. It can then format its screen output based on that country's conventions.

Further Abroad

If you're looking to the European markets, Apple recommends that you concentrate on the United Kingdom, German, and French markets first, then expand to the Swedish and Dutch markets. Many see the U.K. market as the best gateway into Europe. It's about 10 percent of the size of the U.S. market, has many of the same characteristics, and is much less expensive to enter than other European countries. These factors make the United Kingdom a good test market in which to gain your first international experience.

On the other side of the world, Australia is another good place to start building your worldwide market. It's also an English-speaking country and has a market of 18 million people, most of whom live in one of Australia's five major cities. Australians have a high involvement with technology, and Apple products are popular there.

Non-English Markets

Germany and Japan are two other markets worth considering, but they will require more work because of the languages they use. Apple is very popular with Germans, and their taste for consumer electronics—a \$14 billion market there, compared with \$26 billion in the United States—makes it likely that the Newton will be popular there.

Japan is a bigger market—over 123 million people. Like Germany, Japan also has a high acceptance of consumer electronics. (Its consumption of consumer electronics products, at \$24 billion, is only slightly less than that of the United States.) The Japanese Newton MessagePad should be popular with the Japanese because of their attraction to consumer electronics and their familiarity with pen-input devices. And don't think that you'll be locked out of the Japanese Newton market: Currently, 75 percent of the Macintosh products tailored for the Japanese market come from companies outside Japan.

Product Ideas

No Strategy Mosaic column would be complete without some mention of new product opportunities. Here are some ideas—in four categories—from Jeff Henning, of BIS Strategic Decisions.

Personal computer connectivity:

- programs that let a Newton device remotely access and manipulate data from personal computers and networks
- “extractor/injector” programs that “strip” basic data from a personal computer file, let users manipulate it on a Newton device, and then replace the changed data into the “full” computer file, thus allowing users to interact with data stored and maintained on personal computers

Forms-based automation:

- forms-based vertical solutions that solve the problems of a particular client or type of business
- tools that help developers create forms-based solutions

Personal information:

- sales information software, including software that helps non-salespeople perform sales and service tasks
- information navigators, software that helps users search and use large bodies of data stored on the Newton device

Communications:

- software that helps two Newton users in the same room collaborate interactively on a shared document
- automatic response systems that receive requests for information and answer them automatically

StarCore

Apple's PIE management realized that the Newton platform will not succeed if Newton developers, worldwide, can't get their software to the customer and, by doing so, be financially successful. This led to the creation of StarCore, an

international software publishing and distribution group for Newton and CD-ROM products.

StarCore gave several presentations at the Newton conference to acquaint developers with the services it offers. If you arrange to have StarCore publish your product, StarCore will give you guidance in various areas and will take care of the financial, production, marketing, and worldwide distribution of your software; in return, StarCore will pay you a royalty based on your software's sales.

Under the StarCore affiliate label program, you create the finished boxed product; StarCore distributes your product worldwide and helps you market your product. This approach requires more effort and risk on your part but can potentially be more profitable for you.

StarCore is looking for products in six categories:

- business applications
- information and reference
- entertainment
- sports
- edutainment (educational software)
- self-improvement

If you're interested in working with StarCore, you can contact either Jim L'Heureux (for publishing) or Ivy Millman (for the affiliates program). Both can be reached at Apple Computer, Inc., 20525 Mariani Avenue, M/S 305-4A, Cupertino, CA 95014, USA. You can also contact them on AppleLink at addresses LASH and MILLMAN1, respectively.

Like a Rocket

Before the Newton Platform Development Conference, I knew a good deal about the Newton platform and what it's capable of technically, but I could only guess where the Newton *market* is going. Now, I think I know more about where the Newton platform is and where it's headed. I think I'm more confident because I heard the opinions of a lot of people *outside* Apple. Not only are they saying roughly the same things about the Newton—they're staking their businesses on being right.

Even if the Newton platform were frozen today and the Newton MessagePad and Sharp ExpertPad were the first, last, and only Newton products, we'd still have a pretty interesting market to exploit. The Newton is new, the operating system and development environment are both powerful, and developers haven't begun to scratch the surface of the Newton's capabilities. Also, it's a "small" architecture with a short (three to six month) cycle from concept to finished product. Since the Newton is also a new market with no companies in control of a particular niche, the Newton platform, *today*, gives any one or two people with a good idea and a modest amount of money a chance at building a successful business.

But the Newton platform is *not* frozen. Apple will be coming out with new Newton devices. Other companies will come out with Newton devices that are completely different from anything that Apple will ever put out. Two-way wireless messaging will result in applications and services that none of us can imagine today. Just as minicomputers and personal computers made the benefits of computing power available to more and more people, the Newton and wireless

messaging have the potential to make those benefits—and new ones—available almost anywhere.

Already, the Newton MessagePad has won awards, and Apple has formed various technology and marketing partnerships with dozens of companies. (See the news items on page 9 for details.) Behind the scenes, Apple is working on making the Newton faster, improving its handwriting recognition, and bringing the price down. Things are just going to keep getting better and better.

Comparisons to the early days of the Macintosh are inevitable—but think about what that *really* means. Already the Newton has more products, developers, and development tools than the Macintosh did in its first year and a half. (The Newton device is about five months old, and its first developers' conference brought in *over 700 third-party developers*.) If you want to make comparisons, the Macintosh is flying high today, but it took a while to get off the ground—think of the Wright brothers' first airplane evolving into a 747. The Newton, on the other hand, is different. It's taking off like a rocket. ®

Markets for Wireless Communications

One-way messaging Two-way messaging

| | | |
|-------------------------|------------|------------|
| Mobile office workers | 48 million | 14 million |
| Personal communications | 81 million | 10 million |
| Corporate users | 7 million | 3 million |
| Field service | 25 million | 3 million |

Out With the Old, In With the New

Don't panic! We're not redoing *Apple Directions*—not just yet, anyway. What we are doing, as we head into the new year, is making sure that we're on target. We want to be sure, nine issues after its debut, that *Apple Directions* is meeting your needs.

You'll see the *Apple Directions* survey on the facing page; please take the few moments necessary to answer its questions and return it to us. It's our best way of knowing who you are, what you want, and how we're doing. This is your chance to tell us if we're succeeding with our goal of helping you maximize your development dollar.

The survey is also our way of keeping *Apple Directions* fresh, interesting, and useful to you; if your responses tell us that part of the newsletter isn't working, we'll change it. If you want more of a certain kind of content, we'll try to provide it. If certain features aren't of any value to you, we'll eliminate them.

We began getting your feedback with our "trick survey" on AppleLink in September. Thanks to the handful of you who went to the trouble to complete it, and also thanks for your very positive feedback. I'd like to tell you about some of the key results here.

The small number of you who responded to the September survey liked *Apple Directions*, very much. On a scale of 1 to 5, with 1 meaning *not valuable* and 5 meaning *extremely valuable*, you gave us a rating of 4.47.

The section of the newsletter you found most valuable was Technology, which you gave a rating (on the scale just mentioned) of 4.53. News (the front section, which includes Strategy Mosaic and this note) was second at 3.87 and Business & Marketing third at 3.23.

How do we interpret these results? These very preliminary results (preliminary because they're based on such a small number of responses) tell me that *Apple Directions* appears to be doing what it's supposed to. Since 3 is halfway

between 1 and 5, we figure that any rating above 3 is favorable. These results suggest that you find real value in every section of the newsletter.

Similarly, every feature but one received a positive rating. Here are the features and their ratings, again on the same 1-to-5 scale:

| | |
|---------------------|------|
| Human Interface | 4.53 |
| Technology articles | 4.47 |
| Apple News | 4.36 |
| Strategy Mosaic | 4.14 |
| CD Highlights | 3.93 |
| Developer Outlook | 3.60 |
| IndustryWatch | 3.53 |
| Editor's Note | 3.53 |
| Marketing Feature | 3.27 |
| Market Res. Monthly | 3.13 |
| Ask Don Norman | 2.93 |

Do these results reflect your opinion? Please complete and return the survey on the facing page and let us know.

My goal in editing *Apple Directions* is to make every article valuable to you. When I can't do that, I want it at least to be interesting. In addition, I want each issue to contain pleasant surprises that keep you turning the pages and coming back for more. I welcome your survey responses, which will tell me, personally, how I'm doing.

Paul Dreyfus
Editor

The Patent, the Number Two Software Company, and the Superstore Expo

By Amanda Hixson, Consultant, Instant Insight

Well, as predicted in last month's column, we didn't even have to wait for the suits to hit the fan before the U.S. Patent Office decided (all by itself, mind you) that it might have been a bit hasty in granting Compton's New Media rights to just about everything related to interactive multimedia.

In a move previously undertaken only a handful of times, the U.S. Patent Office is going to reexamine the patent it recently granted Compton's. This decision reflects industry-wide feelings that the Patent Office "allowed excessively broad claims by Compton's and ignored evidence of 'prior art'— earlier innovations in multimedia that would render the Compton's patent invalid," according to the *San Jose Mercury News*.

At a minimum, the Patent Office decision to reevaluate will push out any conclusion for at least a year, according to the *Mercury News* article. I would guess that further review will lead to a reversal of at least part of the patent decision. Don't start celebrating yet, but especially if you're in multimedia development, this should be good news.

For more information on this issue, download a copy of the patent from the CompuServe CD-ROM Forum along with a document that graphically explains the interrelationships of the patent's various components.

Who Is That Masked Company ?

If anyone asks you to name the top two software companies in the computer business, most of you will probably not answer correctly. Sure, the top dog is pretty easy to figure out—Microsoft. But the number two company is a little less obvious. No, it isn't Lotus, nor is it Borland, nor WordPerfect. The only other company sharing the multibillion dollar bracket with Microsoft (outside of

hardware vendors such as IBM) is Computer Associates International, Inc., or CA.

Huh? Who?

CA is a little company in Islandia, New York, that did about \$1.8 billion in revenue for fiscal 1993. You might recognize it better as the company that has been giving away software of late. Earlier this year, for example, CA gave away copies of its CA-Simply Money finance package for \$6.95 in shipping and handling charges to the first 1 million credit-card-bearing callers. More recently, it's been giving away copies of CA-Simply Tax, a Windows tax preparation program, for a paltry \$9.95 shipping and handling fee. Unlike the earlier giveaway, there are currently no limits on the number of free versions of CA-Simply Tax the company will give away.

On first blush, CA's giveaways might look like desperate attempts to capture market share. But, when you consider that CA has been around since 1976, pulls down almost \$2 billion a year, and sells products ranging from personal finance programs to mainframe database tools, the desperation angle quickly loses luster. No, the answer is much simpler: CA is using a unique marketing gimmick to gain name recognition and entice new customers into using tools from its broad family of products.

CA is a company with a history of doing smart things, such as buying every second- and third-tier product in every category it could get its hands on. It then integrates them into a product family, which includes revamped versions of SuperCalc and Cricket Draw & Cricket Graph (to name a few).

Its executives realize that giving away financial and tax products isn't a long-term risk. Remember, taxes change every year. Consequently, everyone has to obtain an updated tax program, and, as tax laws change, financial planning tools need to be revamped as well. What CA is doing is akin to the old razor and razor blade trick; give away the razors, then focus on selling the blades.

If nothing else, CA is a fine example of great sales and marketing. Its examples are well worth a look by any company trying to figure out how to survive long-term in the trepidatious computer industry.

Who Needs Another Trade Show ?

If you're like me, you've probably noticed that a new trade show pops up just about every week. Most are simply variations on well-known themes, such as networking, portable computing, hand-held computing, desktop computing, cellular computing, and peripherals, ad infinitum. Rarely do I come across a new show that looks interesting, let alone one that offers a new angle on the old themes.

Imagine my surprise, then, when not long ago, while reading Newsbytes on AppleLink (path—News Break:Apple & Industry News: Newsbytes), I came across a new show that actually looks interesting. This new event is called the *Superstore Expo*, and the five currently planned events for 1994 will have the look and feel of a PC Expo or Macworld with the added attraction of allowing vendors to actively sell products from the show floor.

According to the November Newsbytes article I read, there are already 75 exhibitors, including the likes of IBM, DEC™, WordPerfect, and, of course, Computer Associates (guess what they'll be giving away at the show?). When the article appeared on AppleLink, Apple Computer, Inc., was not listed as a participant, while large resellers such as Merisel were. It seems to me that this would be a good place for you to mix with companies selling products for "other" operating systems, especially with 1994 being the year of Macintosh computers with PowerPC and OpenDoc.

To reduce the risk of running afoul of resellers, the producers of the show are apparently going out of their way to make sure that major exhibitors work with local resellers to establish local follow-up for items purchased at the show. I imagine that most of the folks hawking wares at the show will also make sure they don't undercut the prices their channel partners set for their products. They'll probably send discount seekers to resellers during show hours to ensure

that the channel folks have an opportunity to make a sale. They will if they're smart.

What I like about this whole concept is that it potentially offers folks an opportunity to talk to vendors on product issues while obtaining products on the spot. Additionally, this concept offers lesser-known vendors an opportunity to display their wares in an event anchored by some of the industry's larger players; kind of a traveling, hi-tech shopping mall.

The Newsbytes piece says that the folks sponsoring the show actually produce \$1 million of advertising, 210 radio spots, 115 television commercials, 42 pages of black-and-white advertising in major newspapers with an additional 20 pages of color advertising in Sunday inserts for each of the five shows. If that's true, this should prove to be a tremendous opportunity for those software and hardware vendors who choose to participate.

Sites for the five 1994 shows are San Francisco's Moscone Center in February, Anaheim Center in Los Angeles in May, the Rosemont in Chicago in August, Boston's Bayside Exposition Center in September, and Philadelphia's new Pennsylvania Center in December.

Amanda Hixson is currently a consultant in the area of product and process management. Along with being a five-year Apple alum, during which time she was, among other things, an evangelist, product marketing manager, and software project leader, she is also the author of four books and a successful CAI training tool, a journalist, industry analyst, former book acquisitions editor, accounting manager, and perpetual realist (or cynic, depending on whom you talk to).

***Apple Directions* On Line—March**

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

February 1—Preliminary draft copy

February 15—Final copy

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

Apple PIE in the News

Here's a round-up of recent news from Apple's Personal Interactive Electronics (PIE) division.

German Newton MessagePad Available

On December 17, 1993, Apple Computer, Inc., announced that the German version of the Newton MessagePad is now available from retailers in Germany, Austria, and Switzerland. The German version of the MessagePad is the first localized version of the Newton MessagePad communications assistant to ship in a non-English-speaking area. The flexible design of the German version of the MessagePad enables the handwriting recognition feature and the integrated dictionary to be set specifically for Germany, Austria, and Switzerland. This includes adapting the MessagePad to be familiar with the cultural habits of the country in which it is being used. For example, in English-speaking countries, the week starts on a Sunday, whereas in other countries, the week starts on a Monday. The German version of the Newton MessagePad is fully acquainted with this and other country-specific features.

Launch of Newton Industry Association

On December 7, 1993, the Newton Industry Association was announced at the first International Newton Platform Development Conference, which took place in Santa Clara, California. The launch session of the Newton Industry Association took place the day before, at Apple's R&D campus in Cupertino, California. The participants of the inaugural session included, among others, Alcatel, ARM, BellSouth MobileComm, British Telecom/Cellnet, Cirrus Logic, Deutsche Telecom, GEC Plessey, LSI Logic, Matsushita, Motorola, ParaGraph, Scriptel, Sharp, Siemens/ROLM, Telia, Toshiba, Traveling Software, and US West. These companies represent licensees of Newton technology, OEMs, component suppliers, and marketing partners.

This association is a forum that will promote the growth and compatibility of the Newton platform and associated devices. The association will also promote standards for Newton devices in the areas of wireless communications, telephony support, and office automation. "The active participation in the Newton Industry Association of major corporations from a variety of industries

will contribute to the growing success and the widespread adoption of the Newton platform,” said Gaston Bastiaens, vice president and general manager of Apple’s PIE division.

Recent Newton Awards

At Comdex/Fall in Las Vegas last November, the largest computer industry trade show in the United States, *PC LapTop Computers Magazine* honored the Newton MessagePad with its 1993 Editors’ Choice Award for “Most Promising Portable.” Also at Comdex, *PC Magazine* awarded the Newton MessagePad first place in the “Design Category” of their 10th Annual Technical Excellence Awards, acknowledging the Newton technology and Apple’s sophisticated integration of hardware and software in the MessagePad’s highly portable format.

BYTE magazine recently selected Newton technology, among other Apple products, to receive a 1993 BYTE Award of Excellence. Of the hundreds of new computing products launched in 1993, only 15 received their Award of Excellence. *Reseller Management* magazine identified the Newton MessagePad as one of the “Best-to-Sell Products of the Year” in their 1993 Readers’ Choice Awards. The MessagePad was voted by their high-level executive readership to be the “Most Innovative Product of the Year.”

StarCore Catalog Includes 50 CD-ROM Titles

StarCore, the new software publishing and distribution group formed by Apple Computer, Inc., has announced agreements with InterOptica Publishing Ltd. and the Time Warner Interactive Group to distribute CD-ROM software for Macintosh and Windows platforms worldwide.

The new agreements allow StarCore to distribute more than 40 CD-ROM titles from the two companies under its StarCore affiliate label. With the addition of these titles, Apple’s StarCore group brought more than 50 CD-ROM titles to market in 1993.

Included in InterOptica’s offerings are a series of seven CD-ROM titles co-authored with the Sierra Club on such subjects as the blue whale and Mount Everest; these titles play on both Macintosh and Windows-based personal

computers. StarCore is also distributing dozens of multimedia games, interactive information, and early learning titles published by the Time Warner Interactive Group.

Apple's StarCore group publishes and distributes software on CD-ROM for Macintosh and Windows platforms and on PCMCIA cards and disks for the Newton platform. StarCore distributes both StarCore and affiliate label software through thousands of outlets nationwide, and throughout Europe and the Pacific Rim. StarCore also assists companies working under the StarCore affiliate label program in marketing their products through in-store promotions, catalogs, brochures, and public relations.

U.S. Department of Defense Contract to Explore Newton Technology

Last December 6, Apple Computer, Inc., announced that it has received a \$1 million contract from the U.S. Department of Defense (DoD) to explore the application of Newton technologies within the medical sector of the DoD. In collaboration with KPMG/Peat Marwick, Apple Computer will lead the ProMED project from October 1993 through August 1994 to investigate how the DoD can improve health care services, streamline processes, and lower the cost of health care using Newton technology.

The Concept Introduction phase, which began on November 8th in the three pilot hospitals, focused on the use of Newton as a personal productivity tool through personal organizer functions. Scheduled for implementation in December 1993 through March 1994, the Concept Demonstration phase will introduce applications specific to health care on Newton, such as the ordering of lab and radiology diagnostics, results notification through Newton paging capabilities, and prescription ordering. The third and final phase, Concept Validation, will explore using untethered, bidirectional communication in selected applications (for example, scheduling programs for physicians) and sharing clinical documents (such as problem lists, progress notes, visit/encounter notes, medical histories, and treatment protocols) among the professional staff.

Developer University Offers Newton Training

Apple Developer University is offering a five-day Newton Programming course to interested developers. The dates for upcoming classes, all held in Cupertino California, are

- January 31–February 4
- February 7–11
- March 7–11
- March 21–25

The cost of the course is \$1800. (For comparison purposes, note that this is \$45–\$325 cheaper than all the five-day courses offered by Microsoft NT University.) You can register for these courses by contacting the Developer University Registrar: phone 408-974-4897; fax 408-974-0544; AppleLink DEVUNIV.

Ten Years Ago, Almost Today: Apple Introduces the Macintosh Computer

On January 24, 1984—ten years ago this month—Apple introduced the first Macintosh computer, a “sophisticated, affordably priced personal computer designed for businesspeople, professionals, and students,” according to the official press release from Apple Computer, Inc.

Upon the new computer’s release, *Apple Directions* Technical Editor Gregg Williams, then writing for *BYTE* magazine, called it “the most important development in computers in the last five years.” Apple is on the threshold of making another great technical leap forward with the first Macintosh computers with PowerPC RISC processors. We thought it would be an appropriate time to look back ten years and recall some of the announcement’s highlights in light of the dramatic changes undergone by the personal computer industry since then and to hear more of Gregg Williams’ analysis at the time.

Yesterday and Today

At a cost of \$2,495, the first Macintosh computer shipped with an internal 9-inch monochrome monitor, a keyboard, 128K (yes, that’s kilobytes) of RAM, 64K of ROM, a single floppy disk drive (and no hard drive, although a second floppy drive was an available option), and a Motorola 68000 microprocessor with a clock speed of 7.83 MHz.

Apple already knew from its experience with the Apple II just how significant delivering great software and other products would be to making the new platform a success. The first versions of MacWrite® and MacPaint® (which were almost called *Macauthor* and *Mackelangelo!*) were available free to the first Macintosh customers. Apple had already partnered with developers to help them produce software products for the new platform. On January 24, 1983, 100 companies announced their commitment to developing Macintosh products.

Apple said that it anticipated sales of 350,000 units the first year; 50,000 Macintosh computers had sold 74 days after the introduction, a sales rate nearly

three times as fast as that of the first IBM PC, which took seven months to reach the 50,000 mark.

“My, how things have changed” is probably the understatement of the year. Today, \$2,500 will buy the Macintosh Quadra 610, with color monitor and keyboard, 8 MB of memory, 512K of VRAM, a 230 MB hard drive, the Motorola 68040 microprocessor running at 25 MHz, and an internal CD-ROM drive.

There’s no question how important your contribution, and the work of other developers, has been toward making the Macintosh computer the success it’s been.

Today, there are so many Macintosh products available from non-Apple third parties that Apple no longer tracks the actual numbers. However, Apple estimates that there are currently more than 10,000 Macintosh applications available in the English language, with at least another 6,000 European titles. (Estimates for Asian-language applications weren’t available.)

Also, currently, 10,000 developers worldwide are enrolled in Apple’s Macintosh developer programs—6,000 of those in the United States—and APDA numbers its customer base for Macintosh development tools at 30,000.

How about unit sales figures? Steve Jobs and the original Macintosh team couldn’t have dreamed of what was to happen to the personal computer market. Last year, Apple sold 50,000 Macintosh computers worldwide approximately every *five days*; it currently takes just over a month to sell 350,000 Macintosh computers. (For a complete report on worldwide and U.S. Macintosh sales in 1993, see the next news item.)

“Superb American Technology”

Apple Directions wasn’t even a gleam in anyone’s eye ten years ago, but our technical editor, Gregg Williams, was already covering the personal computer beat. Writing in the February 1984 issue of *BYTE*, he called the first Macintosh computer “a superb example of what American technology can do when given the chance.”

Gregg predicted that the Macintosh computer would have “three important effects.” First, he said that the computer would be “imitated but not copied . . . but those companies that try to imitate the Mac on other machines will have trouble matching its price/performance combination.” Gregg was only half right; as yet, there is still no effective copy of the Macintosh computer, and many would argue that the combination of features and ease of use offered by the Macintosh has always made it a superior computer at its price.

However, no one was to know at the time how great an issue price/performance expressed mostly in terms of RAM and microprocessor clock speed would become over the next decade. In fact, it’s only since the release of Apple’s fall 1993 products that the price and performance of the Macintosh computer have begun to pull ahead of the competition again.

Gregg’s second prediction couldn’t have been more accurate: that the 3.5-inch floppy disks used by the first Macintosh computers would become the “magnetic medium of choice” for personal computer manufacturers and users instead of the larger floppy disks popular at the time.

Third, Gregg thought that the Macintosh computer would “increase Apple’s reputation in the marketplace” and “delay IBM’s domination of the personal computer market.” Here, he was mostly right; even with its relatively small share of the marketplace, Apple has continued to wield significant influence in the personal computer industry. And that influence has prevented the “other” platform from walking away with the personal computer industry lock, stock, and barrel.

That’s *Macintosh*, not *McIntosh*

The Macintosh computer has been influential in another, completely unanticipated way. Apple first code-named the project *McIntosh* after the apple. In early testing, users began misspelling the name as *Macintosh*, and the appellation stuck. The other day, Gregg Williams saw a sign in a local supermarket advertising Macintosh apples.

Apple Worldwide and U.S. Unit Shipments Increase 30 Percent in 1993

International Data Corporation (IDC) recently released its projections for personal computer shipments in 1993, and the data contained signs that Apple's aggressive market share strategy is working, helping to give you a larger customer base for your Macintosh products.

Apple shipped 3.6 million Macintosh computers worldwide in calendar year 1993. That's 10 percent of all personal computers shipped last year and 30 percent more than Apple shipped in 1992, according to IDC's projections, which are based on three quarters of actual shipment figures.

In the United States, the world's largest personal computer market, Apple shipped 2.05 million units, 13.9 percent of total U.S. personal computer shipments and 32 percent more than 1992 shipments.

Apple's worldwide and U.S. shipments were second only to IBM's; a year ago, IDC numbers gave Apple first place in the 1992 U.S. unit-shipment sweepstakes, narrowly ahead of IBM. This year, IBM finished only 25,000 units ahead of Apple in the United States. Worldwide, IBM shipped 4.4 million personal computers.

Total personal computer shipments around the world increased 19 percent, according to IDC estimates; together, the world's personal computer vendors shipped 36.1 million units in 1993 compared with 30.4 million in 1992. In the United States, 14.8 million personal computers shipped in 1993, 11.8 million in 1992—an annual increase of 26 percent.

The table below "Personal Computer Shipments, 1993" shows IDC estimates of the number of units shipped by the top ten personal computer vendors worldwide and in the United States.

Personal Computer Shipments, 1993

Worldwide

| '93 rank | '92 rank | Vendor | Projected unit shipments | 1993 % share | 1992 unit shipments | Actual % share | Growth 93/92 |
|-------------|----------|--------------|--------------------------|--------------|---------------------|----------------|--------------|
| 1 | 1 | IBM | 4,400,000 | 12.2 % | 3,210,000 | 10.6 % | 37 % |
| 2 | 2 | Apple | 3,600,000 | 10.0 | 2,760,000 | 9.1 | 30 |
| 3 | 3 | Compaq | 3,045,000 | 8.4 | 1,555,000 | 5.1 | 96 |
| 4 | 5 | NEC | 1,541,000 | 4.3 | 1,376,000 | 4.5 | 12 |
| 5 | 7 | Dell | 1,188,000 | 3.3 | 670,000 | 2.2 | 77 |
| 6 | 6 | Packard Bell | 1,100,000 | 3.0 | 685,000 | 2.3 | 61 |
| 7 | 9 | AST | 944,000 | 2.6 | 594,000 | 2.0 | 59 |
| 8 | 4 | Commodore | 937,000 | 2.6 | 1,445,000 | 4.8 | -35 |
| 9 | 8 | Toshiba | 750,000 | 2.1 | 630,000 | 2.1 | 19 |
| 10 | 12 | Gateway 2000 | 670,000 | 1.9 | 437,000 | 1.4 | 53 |
| All vendors | | | 36,101,000 | | 30,383,000 | 19 % | |

United States

| '93 rank | '92 rank | Vendor | Projected unit shipments | 1993 % share | 1992 unit shipments | Actual % share | Growth 93/92 |
|----------|----------|--------------|--------------------------|--------------|---------------------|----------------|--------------|
| 1 | 2 | IBM | 2,075,000 | 14.0 % | 1,374,600 | 11.7 % | 51 % |
| 2 | 1 | Apple | 2,050,000 | 13.9 | 1,550,000 | 13.2 | 32 |
| 3 | 3 | Compaq | 1,418,000 | 9.6 | 675,820 | 5.7 | 110 |
| 4 | 4 | Packard Bell | 997,000 | 6.7 | 624,000 | 5.3 | 60 |
| 5 | 5 | Dell | 795,000 | 5.4 | 440,000 | 3.7 | 81 |
| 6 | 6 | Gateway 2000 | 644,000 | 4.4 | 428,180 | 3.6 | 50 |
| 7 | 8 | AST | 530,000 | 3.6 | 320,000 | 2.7 | 6 |
| 8 | 7 | Tandy | 350,000 | 2.4 | 329,870 | 2.8 | 6 |
| 9 | 9 | Toshiba | 316,000 | 2.1 | 242,950 | 2.1 | 30 |
| 10 | 16 | ZDS | 307,000 | 2.1 | 165,000 | 1.4 | 86 |

| | | | |
|-------------|------------|------------|------|
| All vendors | 14,800,000 | 11,761,423 | 26 % |
|-------------|------------|------------|------|

Tool Chest Edition, February 1994: *In the Seek of the Night*

Welcome to the February Tool Chest Edition of the Developer CD. This month brings a newly organized Testing & Debugging folder with several new utilities, thanks to Apple summer intern Kelsey Schwind. Also, a late beta version of the new Contents Catalog continues to lurk in the CD Info folder, awaiting your testing and feedback. Here are some of this month's new features.

ABS Technical Notes Updates

Beginning with the March 1994 Reference Library Edition of the Developer CD, the ESD Technical Notes folder will be renamed ABS Technical Notes. This folder contains new and revised ABS Tech Notes for AppleSearch, Apple Workgroup Server products, and Data Access Language.

AppleSearch Developer Info

This package provides developers with information on how to create or develop products using AppleSearch, as well as those to be used with AppleSearch. It contains two documents:

- AppleSearch Developer's Kit Overview
- AppleSearch Update File Format Reference

Developers need to sign the AppleSearch Developer's License for using and distributing AppleSearch.

Chinese Language Kit

The Chinese Language Kit contains all the software you need for working in Chinese on your computer. With it, you can use a Chinese application program, or a program that takes advantage of WorldScript, to enter Chinese and other languages you work with in a single document. The Chinese Language Kit supports both Traditional Chinese and Simplified Chinese scripts. You can install and use one or both of them.

Developer Notes Update

Developer Notes feature descriptions of new hardware and software features, comparisons with existing CPUs, and information about expansion card design. This month's new note covers the Personal LaserWriter 320 printer, a low-cost, 300 dpi version of the Personal LaserWriter 300, incorporating PostScript™ II. Hard-copy versions of Developer Notes are also available through APDA.

Inline Input for TextEdit

This package helps developers implement inline input for TextEdit using the TSMTE extension in their applications. The package provides sample code for implementing inline input for TextEdit, the TSMTE interfaces, and the TSMTE extension. It is intended to be used along with Macintosh Technical Note TE 27: Inline Input for TextEdit with TSMTE.

Mac Tech Notes Updates

Technical Notes are a collection of short (and not-so-short) articles dealing with specific development topics. New and updated Tech Notes for February 1994 include ME 14: New Memory Manager and OV 20: Internationalization, PR 10: Printing Loop . . . Cares.

SourceObj

SourceObj reads a .c.o file and its corresponding .c file and produces a mixed listing of 680x0 assembly and C source code as output. It helps you to write better C and Pascal code by providing a way to view the output of MPW 680x0 compilers on a per-source-line basis. See the file About SourceObj for details.

Universal Interfaces

With the impending introduction of PowerPC processor-based Macintosh computers, it's important that your applications use portable source code so they may be easily ported to run in native PowerPC mode. In addition to portable source code, Apple Computer, Inc., now requires header files to work for several different compilers. These files are called *universal CIncludes*.

The concept behind these universal interface files is that the same file can be used for any compiler on either the 680x0 or PowerPC platform without changes to the interfaces and, consequently, to your source code.

Alex Doshier
Developer CD Leader

The Story Behind the Power (Macintosh With PowerPC, That Is), Part One

An Interview With Sheila Brady
and Jim Gable

Apple Directions first began telling you about the significance of the PowerPC processor-based Macintosh computer almost two years ago. Since then, the key message has been “Go native!” Apple Computer, Inc., wants you to port your existing applications, written for CISC (complex instruction set computing) 680x0 processor-based Macintosh computers so that they can take advantage of the performance and features of the future RISC (reduced instruction set computing) Macintosh computers with PowerPC.

With the release of Apple’s Macintosh on RISC Software Developer’s Kit at Macworld in January (see page 1 news story), you can now get started porting your products to the new platform in earnest. You’ll want to start porting soon: Release of the first Macintosh computers with PowerPC is on schedule for the first half of 1994. Your applications running in native mode on the new computers are expected to run two to four times faster than they would on the highest performance 68040-based Macintosh systems and Intel x486-based systems.

To give you technical and marketing details about the soon-to-be-shipped RISC Macintosh computers, *Apple Directions* recently met with two of the leaders of the Macintosh with PowerPC project at Apple, Sheila Brady and Jim Gable. According to legend, and most reliable witnesses, the System 7 project foundered until Sheila was appointed its leader and took the necessary steps to deliver the new version of the Macintosh operating system. Sheila, an eight-year Apple veteran, is now the project leader for the software half of the Macintosh with PowerPC project.

Jim is PowerPC product line manager; he’s also been with Apple for eight years, and during his tenure he has been product marketing manager for a variety of LaserWriter and ImageWriter products as well as the TrueType project.

The license plates on his car say “MAC RISC,” which can only mean that Jim drives two to four times faster than the rest of us.

Here is the first half of our conversation with Sheila and Jim, in which we covered a variety of technical issues.

...

Apple Directions (AD): *What was the original concept behind putting RISC processors into the Macintosh computer? Was it simply a matter of boosting performance?*

Sheila Brady (SB): Well, that, and also providing a foundation for the future. The Motorola 680x0 line just seemed to be running out of steam, and not only in terms of performance. RISC just seemed the wave of the future, CISC something that would be left behind.

Jim Gable (JG): True, true. We were considering other approaches, both on the hardware and the software side, but they evolved into the PowerPC program a couple of years ago. We've had some engineers looking at RISC for a long time, and the microprocessor experts felt it was inevitable, that if we didn't switch to RISC, we'd be in trouble.

SB: We'd be out!

JG: And those are the guys who set the early foundations for the Macintosh with PowerPC project. But we wouldn't have the product we're going to release if it

weren't for the software folks. By comparison, the hardware side of things looks easy now.

The reason the DOS/Windows world isn't going to RISC processors is that Intel is so dominant. But workstations are all RISC, and personal digital assistants are primarily RISC. Anybody introducing a new platform today goes for RISC. It's just a better architecture for computer systems as we head into the future.

Project History

AD: *The first Macintosh computer with PowerPC is very close to shipping. The two of you have probably had as much to do with that achievement as anybody, probably more than most. We'd like to hear about how Apple went from the initial idea—of using RISC processors in the Macintosh computer—to the actual products.*

SB: Jim is the marketing guru, really the leader of the program. He's been the person who's been shepherding it through, right from the early days—the visionary who's been making everything happen and the person who's been out there selling it all the way along. He's beating all the engineers into actually building what he's been imagining that we should build.

This is another project where we tried to look at the complexity of putting together software and hardware—realizing that the big challenge is compatibility, measuring that against the need for high performance, and trying to keep all the engineers and their management on the straight and narrow, to focus, focus, focus. . . .

JG: As usual, Sheila is completing misstating this! There would not be Macintosh with PowerPC if Sheila weren't on the project, because before Sheila

agreed to do this project, it was . . . completely chaotic. Different people had different ideas of what they wanted to do with it. We couldn't get attention. We couldn't get focus.

The hardware guys were kind of out starting work, but they were starting work with no software, no firm plans and it was really not a good situation. That was a while ago and I remember it well!

The Bandwagon

AD: *This was, what, a year and a half ago?*

SB: I took the job in the beginning of January, so it's exactly two years ago, amazingly!

JG: The key assignments that got the software side kicked off were Sheila and Phil Koch. Phil had such a good reputation, coming in from Dartmouth, of being a real expert on the Macintosh at a technical level. People really felt that the down, dirty, low-level technical things would be done right. By all measures, so far, it looks like that has happened. Probably more important, when people at Apple saw that Sheila was going to do this, they all of a sudden said, "Oh, heck! This must be important!"

Now it's obvious that PowerPC is important, because people can see software running two to ten times faster. They can see developers very excited. They can see customers really looking forward to the product. Now everybody wants to get their project working on PowerPC. Now everybody is on the bandwagon, which is great. What really will make the difference are the people who jumped on two years ago and drove the project to where it is today.

SB: The big difficulty at the time was that there was this big thrash going on about which RISC chip we should actually pick, and then we went and picked the IBM chip, and, afterward, there was a lot of debate about whether or not that was the right move.

AD: *IBM's Power chip?*

SB: Yes, the PowerPC chip that was based on IBM's Power architecture. I think it was around that time when people realized, "Well, the argument is over. You're not going to win if you keep on arguing. It's done. This is the chip we're going to use." Everybody soon got an attitude of "Sit down, shut up, decide to be on the program or off it."

And at the same time, the thing that was really good was, as Jim said, getting Phil Koch to commit and say, "All right, let's map out an architecture that we can actually deliver for running the assembly language of a 68000 microprocessor on a PowerPC processor-based machine."

JG: The work that Phil's software team has done is going to be more important than people anticipate, because when you first turn on a Macintosh computer with PowerPC, it acts like a Mac. It's very compatible. Things work the way you expect them to. And like a pro athlete, Phil made it look easy.

And Macintosh with PowerPC is kind of like that pro athlete, too. You turn it on and it looks simple. But underneath, there's a dramatic amount of new software going on. The entire memory architecture is different. The way we use virtual memory is different. A new manager, the Code Fragment Manager, allows us to have shared libraries in a very smooth, simple, easy-to-use way that we've never

had before. All this is going on at a level that users never see and, in fact, developers, if they choose to, never have to touch.

Compatibility and the Emulator

AD: *How has the project coalesced at Apple since those early days two years ago?*

SB: Actually, it started almost three years ago, a year before I got involved. At that point, we had a “Let’s dive into RISC!” kind of program, and then some people left the company and others decided, “Well, that’s not really the right thing to do.” In the meantime, there was a program going on in the background with people wondering, “What if we just did the simplest possible thing? If we had a Macintosh and we just yanked out the 68000 chip and we plugged in a RISC processor?”

If you take a look at it, it’s a very simple concept; it’s very easy to carry in your mind. It’s not like lots of new, different types of buses or a new graphics or I/O subsystem. It’s just, take the CISC chip out, replace it with the RISC chip, and move on from there.

When they first had that idea, Gary Davidian began writing the emulator that lets 68000-based applications work on the RISC chip. He was absolutely brilliant. He wrote the basis of the emulator for another RISC chip we were considering at the time; then he essentially ported it over to work with the PowerPC processor.

JG: And it’s very interesting how the technical vision of a few people not only ended up being right, but also was able to disprove the nay-sayers. Apple is a big enough place, now, that if anyone has an idea, there are a lot of bright

people to tell you why it won't work. People were saying, "That won't work because of timing dependencies, because of strange 68000 behaviors." There were all these reasons why the emulator could not work. So Gary decided to prove that they were wrong, and he wrote an emulator that cannot be broken.

It's really quite astonishing. The end product is extremely compatible. There are some compatibility problems, because whenever we do a new Macintosh, there are things that cause old software to break—but it's for reasons other than the emulator. To date, we know of no compatibility issue that is caused by the emulator.

Taking System
Software Native

AD: *So, in other words, software that breaks on the new RISC computers would very likely break on any new Macintosh model that uses a 680x0 processor?*

JG: Precisely. For example, with our recently introduced AV computers, the Macintosh Quadra 660AV and Quadra 880AV computers, some things break on them, and usually those same things break on Macintosh with PowerPC computers, because we've borrowed a lot of the AV architecture.

SB: When we talked to customers in focus groups at the beginning of 1993, what we heard was that compatibility was absolutely critical. This was very convenient for us, because it totally dovetailed into our technical strategy, the first part of which amounted to proving that the emulator technology actually works. We did that in the very first prototypes. We moved on from there, saying, "OK, now we have this situation where we have emulation working, and we have the PowerPC processor available to our system software. Let's see how much of the system software we can actually take native on the product."

So we started moving parts of System 7 native. As we moved them into native mode, we started seeing slight compatibility problems, but performance went up tremendously, because we were running more and more of the Toolbox native.

There were two factions in the program. There was one set of people who said, "Just put the emulator on the product. Period. That's all you need to do." And then another faction essentially said, "No, no, we really want to add the complexity of RISC to the program and provide a mixed-mode architecture, to be able to run either the native PowerPC code or the 680x0 code."

We ended up taking a bit of a risk in going ahead and doing that, but it's been worthwhile.

JG: Right. The mixed mode is what makes the machine work.

AD: *Tell us more about the mixed-mode architecture.*

JG: It's a new part of the operating system that allows code to switch back and forth between 680x0 code streams and PowerPC streams. It's constantly switching as you're executing an application. It's really the whole secret behind acceleration. Not only do we deliver acceleration at introduction, but we have a mixed-mode architecture in which we didn't have to port the entire operating system. That would have taken a long time, broken a lot more applications, and been a really huge project. But we didn't have to because of the mixed-mode concept.

AD: *That switching takes place transparent to the software?*

SB: Completely, completely. One of the challenges for us in improving performance while maintaining compatibility has been to try to figure out the correct routines to take native. There are a number of different boundaries that we've looked at.

Obviously, you want to look for those routines that are executed really frequently by the application. But sometimes, some of those routines are extremely tiny, and if you have a routine that's only three instructions long and the context switch for mixed-mode is, say, 100 instructions, well, then, it creates a lot of overhead to actually move execution back and forth. So we had to do a little bit of analysis to really figure out which routines to port to PowerPC.

For routines that are only three instructions long, we've provided what we call "fat binaries," so they're available in 680x0 format and they're also available in PowerPC format.

We also had to look at the entire call chain. Say you had a routine translated into native mode, but within it, it made a couple of calls to something that moved around memory that was maybe a really tiny 680x0 routine. Well, you'd end up being burdened by the full context switch that would have to happen for the tiny routines. So we've had to go all the way down through certain call chain sequences and make sure that every single different routine is actually translated to native.

When we started doing that, it was really a fun time to be part of the project. Suddenly, people discovered, "Oh, this is the place!" You know, someone ported a minor routine and then all of a sudden we'd see a performance improvement, boom!, because we'd modified that one, unique, little routine.

Native Routines
and Managers

AD: *Can you give a specific example of that?*

SB: HLock, I think, was one; when we took it native people went, “Oh, my God! HLock!” It’s a routine that just locks things in memory. We found that in the middle of putting together Diatom, the demo program we used to show off the performance of PowerPC. That was the one that actually set Diatom off to be as impressive as it turned out.

AD: *What was at stake in translating HLock to run in native mode?*

SB: Basically, it’s something that’s called many, many times, because it locks blocks in memory; then there’s HUnlock, which unlocks them. As you go down through a routine that’s reasonably complex, it makes a call to HLock. If everything else in the call chain is in native mode, the computer executes on the PowerPC processor through the entire call. But then it’ll hit this HLock. If it is still in 680x0 assembly, it has to carry the burden of an entire context switch, which might take place in the middle of a loop that you actually want to run very fast. So you suddenly lose the advantage of native PowerPC performance by having a context switch happen, get back over into 680x0 mode, execute the HLock, then context-switch back to PowerPC mode to execute the native code again.

If this is in the middle of a loop, you end up running into this performance hurdle that has to happen over and over and over again. When we took HLock native, we didn’t have to have that context switch, and suddenly all the routines that use HLock could take full advantage of native performance.

AD: *What did you do to take HLock native?*

SB: Oh, we just rewrote it. It was in 680x0 assembly language and we rewrote it in PowerPC assembly. That's just one of the routines that isn't ported to C yet. At least I don't think so.

AD: *How many routines have you ported to native mode so far?*

SB: About 10 to 15 percent of the code in System 7 has been ported to native mode and we think that covers almost 90 percent of the execution paths run in the processor. If you think about it, there's a ton of routines that are used at initialization, at startup, that set up all of the data structures, memory, and whatnot. You really don't need to execute these as rapidly as you do a lot of other routines, especially graphics and memory-management routines. Those are the ones we focused on trying to make as pretty as possible—the routines that really make a difference to performance.

AD: *On a much higher lever, say the manager level, what parts of the system actually run in native PowerPC mode?*

SB: Let's see. QuickDraw, the Memory Manager, QuickTime, the emulator, a lot of the Code Fragment Manager. The Script Manager is going to be almost entirely native, as well as a bunch of pieces of TextEdit. A bunch of the Font Manager is native, as well.

System 7.1.2

AD: *Can you say something about the first version of System 7 for Macintosh computers with PowerPC? What are its characteristics?*

SB: It's just like System 7, only it's called System 7.1.2. Everything that's new and interesting is under the hood. For developers, the Code Fragment Manager is probably the most interesting new feature besides the port-routine descriptors that are specifically for native code. As far as the user interface, it's exactly the same.

AD: *So it's basically System 7.1 for the PowerPC processor?*

JG: That's precisely what it is. In fact, developers, if they choose to, can take their code, make a few minor adjustments, and just recompile it and run. They don't even have to look at the new features if they don't want to. In fact, many of the first native applications are built that way. They simply run as they used to, but they're recompiled to run native so they get the performance boost.

AD: *That's assuming that the program was first written in C or C++.*

JG: Even people who are translating from 680x0 assembly language to PowerPC assembly language aren't necessarily taking advantage of the Code Fragment Manager or other new aspects of our RISC system. They could if they wanted to, but the interesting point is that developers, if they want to, don't have

to “rearchitect” their applications at all. They can save that for a future revision. For the first PowerPC version of a product, to get something on the market quickly, they can just recompile.

The Code

Fragment Manager

AD: *We’ve never told Apple Directions readers about the Code Fragment Manager. What can you tell us about it?*

JG: The Code Fragment Manager enables us to offer shared libraries, so-called dynamically linked libraries (DLLs), on Macintosh computers with PowerPC; it’s also being developed for 680x0-based computers. It means that all PowerPC systems from Apple will have, from the beginning, the ability to offer DLLs to developers. We expect, over time, that this will make systems more robust.

It’s one of these things that I think is really forward looking. Today, developers have to do their own thing to share code, which can be unreliable and inconsistent from product to product. That can be confusing for the customer. But with DLLs taken care of by the Code Fragment Manager, it’s a very straightforward process. The capability is built in at the operating-system level.

AD: *What benefits does the Code Fragment Manager have?*

JG: There are several. If developers use it for their own code, it doesn’t reduce code size, per se. But it’s a way of dynamically loading code at some different part of an application’s execution instead of having to load it all at once when the user starts up the application. There are some operations that aren’t commonly used by programs, and you’d rather not have every user’s system burdened with

that code, so you can put it in a library that's shared by various applications and only loaded when it's needed.

AD: *So it's present on a user's hard disk, but it's only loaded into RAM when a user chooses a menu command that uses it?*

JG: That's true, but it's not only a way of doing memory management. You also use it to share common code between multiple applications. For example, in a family of products there might be certain libraries, say a dictionary or spell-checker, that all those products need. Today, you might duplicate it in every application, which has a variety of drawbacks. But with the Code Fragment Manager, you could make it into a DLL, which would be much simpler for you and for the user. In the future, Apple will be able to release new system extensions as DLLs.

SB: Hopefully, we'll be able to get to the point where users will be able to decide if they want a particular feature. They can just drop the shared library into the appropriate folder, the System Folder or wherever, and, without restarting the computer, they immediately see the new menu commands or whatever it is they want. It's great foundation software for the future.

JG: It could also improve the INIT and cdev difficulties we've imposed upon our customers. Today, the biggest source of Macintosh instability from a customer standpoint generally consists of INITs and cdevs. They're wonderful and we all like them, but when you load up your system with them, they not only use more memory than most people realize, but they often conflict with each other. There's not really an established pattern about how you're supposed to build INITs and cdevs and make them so they don't clash with each other.

Now, developers can take some of those functions, at least, and make them DLLs instead of INITs and cdevs. That would be a step toward better system stability, which we hear from customers would be a terrific thing.

AD: *I understand it's also a vital part of OpenDoc?*

SB: Oh, yeah, absolutely!

JG: It's important for OpenDoc, because essentially every OpenDoc part will have the characteristics of a DLL—a piece of code that's not called into RAM until the user requires its features. The Code Fragment Manager can handle linking of OpenDoc parts from the operating system. *[Editor's note: A future issue of Apple Directions will include an in-depth article on Apple's DLL strategy and solutions.]*

Porting: It's Easier
Than Many Think

AD: *Shifting to another subject, I've heard that porting applications to run native in PowerPC is easier than many people expect.*

SB: Yes. Way easier.

JG: Yes. Developers generally think it's going to be a fair amount of work until they actually sit down to do it.

AD: *So, just how easy is it?*

JG: It varies, obviously, but we've had developers come in to Apple to work in our evangelism labs, and they walked out in a couple days with an initial port of their software, a working, native version of their application.

SB: I think Fractal Design's experience with its Painter application was pretty par for the course. It's very complex software. It is all written in C, and it took a total of five days for Fractal's engineers to go from not even understanding what the porting process was to having a preliminary PowerPC version up and running.

JG: Now that's for somebody whose code is written in C or C++. Obviously, that's the smoothest path; but with third-party tools that will soon be available, we'll have solutions for other languages like FORTRAN and Pascal. There are tools to help you convert Pascal to C if you want to do that. There are tools to help you convert even assembly to assembly, if you want to do that. So even if developers aren't in a situation where they have the simplest solution, there are still alternatives to take. Some people are being very successful with them.

[Editor's note: See "PowerPC Development Tools Announced" on page 5 of the October 1993 issue of Apple Directions for a listing of third-party PowerPC tools.]

Testing and Debugging

AD: *Are testing and debugging any more difficult for the Macintosh with PowerPC than they are for the 680x0 computers?*

JG: Once you get to the debugging and testing stage, it's pretty much like it would be for 680x0 applications. There is nothing enormously difficult; it takes approximately the same amount of time, except in one regard it's simpler. This is because there's not such a big family of products to test software with. With an application for 680x0 Macintosh computers, you need to test it way back, even as far back as the Macintosh Plus.

SB: After porting, people have found that software is pretty much bug-for-bug compatible. One guy said, "I kept hoping these bugs would go away. But they didn't, even with the new compiler!" We told him, "Well, they're in your code, buddy!"

Off to a Fast Start

AD: *What else would you like to tell developers about the technical aspects of preparing PowerPC products?*

JG: It's important to recognize that a number of developers are on a roll now with products for the Macintosh with PowerPC platform; it's not going to be a slow start. We're planning to see a number of applications released the day we introduce the first computers. We believe that in 1994 you'll be reading about new native applications every week. People are really getting going, especially now that the tools are available to everyone.

AD: *Can you tell us just how many developers will have products ready on the date of introduction? How about the rest of the year?*

JG: Our goal was to have five native applications ready on the day of introduction. We know we're going to have more than that. We originally expected to have 20 a month after that, but we're also going to exceed that number.

SB: Then, in the first year, we were hoping we'd have 100 native applications, and we'll probably get many more than that.

We know that the computers by themselves mean nothing. What sells the computers are the applications, and we've spent a tremendous amount of energy getting the tools ready so developers can take their applications native and take advantage of the new performance and the new market. ®

Next month: In part two, Sheila Brady and Jim Gable discuss the market Apple expects to reach with the first Macintosh with PowerPC computers and a variety of other issues.

A Few of My Favorite Things

By Pete Bickford

Perhaps it's all the holiday cheer I absorbed at the end of last year, but somehow I feel compelled to dispense with the usual interface kvetching and try to start off the year on a positive note. For a change, I'd like to say some nice things about human interface design.

Programmers work day and night at a seemingly thankless task. They tolerate incompatible hardware, ridiculous schedules, and the fantastic projections of marketing types who have never dereferenced a handle in their lives. Then, some interface geek comes along and tells them their icons aren't pretty enough.

Despite all this, some programmers manage to create truly great programs, with innovative features and great interfaces. For once, I'd like to devote a column not to beating up bad interface design, but instead to recognizing some of the ones that were done right. While none of these applications may be "perfect," each one carries a powerful lesson about good interface design.

Adobe Photoshop—Making Novices Look Good

When I first played with Adobe™ Photoshop several years ago, I did all the standard things that people too busy to read the manual do: I tried out each menu command to see what it did, played around with the paint tools, and had great fun distorting, coloring, and adding special image effects to various graphics. By using the default settings of the various tools and filters, even a non-artist like myself could get sophisticated-looking results.

As the years went by, my graphics needs became more and more advanced. Suddenly, I "discovered" that my favorite easy-to-use painting program was also a high-end color-retouching application. It never hid these advanced features from me, but they never got in my way, either. As my graphic needs have moved from 72 dpi bitmaps printed on my ImageWriter II to huge four-color Linotype jobs, Photoshop has always been my tool of choice. It's an extraordinarily "deep" program with the ability to control virtually every aspect on the image being

edited; at the same time, the designers set it up in such a way that novices could “grow into” the program, getting pleasant-looking results using the default settings, then being able to tweak things to the nth degree as the need arose.

MacWrite Pro—Elegance and Attention to Detail

I waited a long time for this one (a really long time), but when it finally shipped, it didn't disappoint. Claris® MacWrite Pro somehow managed the near-impossible trick of adding great power to its old version while retaining a simple, streamlined feel. The “Pro” version of MacWrite adds style sheets, text and graphic frames (like a page-layout program), tables, and other advanced features. At the same time, its interface remains elegant and understated. It gives you all the power you want without distracting you from the fundamental business of writing.

In addition to its overall elegance, MacWrite Pro is remarkable for its attention to detail. It's obvious that the developers worked very hard to get the little things right—from the artistic layout of their dialogs to the way command keys work in dialog boxes.

While much of the competition becomes more bloated and cluttered with each successive version, MacWrite Pro serves as a great example of a new release done right, at least to me.

Help!—Taking the Terror Out of Errors

Extension conflicts, software incompatibilities, configuration problems—just about every computer owner has faced these at one time or another. My mother, in particular, used to call me every two weeks from Denver to find out why the 1985 version of GlitchInit 1.2 was crashing her Macintosh Quadra.

Help! may be responsible for a great reduction in her long distance bill. It's a very simple program that produces a gorgeous, easy-to-read report of all the potential software problems in your system. In nonthreatening language it identifies the software causing the problem, tells how to fix the problem, and even gives you the number to call to get a new version of any outdated applications you may have.

By using a constantly updated “knowledge base” of rules to supply its intelligence, Help! makes the rather complex job of diagnosing a system seem simple and straightforward. What’s more, the language it uses to point out problems is a great example for anyone who needs to write error messages: It’s clear, concise, and informative. It tells the user exactly what the problem is and how to solve it without using technobabble or talking down.

Norton Utilities—Keeping Problems From Being the User’s Problem

Norton is another of those indispensable utilities that solves the most complex and obscure problems without being complex and obscure itself. Few things in life are as involved as repairing random damage to a hard disk, yet Norton’s Disk Doctor lets even the most casual computer user accomplish this crucial task with confidence.

In another area, Norton Utilities serves as a useful model for the way large business systems should be structured in order to give the various modules the sense of being part of a greater whole. Norton Utilities uses a “launching pad” of icons to represent the various applications that make up the package (for example, Disk Doctor and Speed Disk). Clicking any of these icons takes you to the appropriate application, giving you the appropriate menu bar and so on. When you exit that application, however, you go back to the main “launching pad.”

As a result, Norton Utilities comes across as an integrated system, even though it may actually consist of multiple applications. A similar technique can be used in developing business systems to give the user a sense of overall mastery of the system, while giving the developer the freedom to separate the various modules (such as Purchasing and Shop Floor Control) into separate applications that make use of their own specialized menus.

TouchBase 2.0—Making Life Easier for the User

TouchBase 2.0 is a personal contacts manager that has long been one of my favorite examples of a program that acts intelligently to speed data entry. Quite simply, TouchBase knows that city names are capitalized and state abbreviations consist of two capital letters, and it knows how American phone

numbers are formatted. So, in fields where it expects telephone numbers, it knows that if you enter something like “4085551234” it should really be formatted as “(408) 555-1234”.

Furthermore, since its phone dialer knows your home area code, TouchBase knows that if you enter just the last seven digits (for example, “555-9876”) the number is probably local, and the full number is “(408) 555-9876”. You can even enter strings like “555-4567 Work,” and TouchBase knows enough to format the numeric part correctly.

Perhaps just as important, it knows when not to apply its formatting, such as with extended phone numbers or special dialing codes, or when you specifically override it. TouchBase 2.0 is also notable for its clean, elegant interface and the obvious care that went into the details of its design.

SoftPolish—A Power Tool for Getting the Details Right

An interface design consists of two parts: the grand design and the pesky details. SoftPolish is a developer’s tool that gets the second part right.

SoftPolish reads through your application’s resources to find spelling errors, improperly sized dialog items, misused command-key equivalents, badly masked icons, and many more of the “mechanical” problems of human interface implementation. It also provides programmers with a collection of useful resource validity checks.

Just as any decent SQA environment uses various tools to check over program code, it seems that we programmers would be remiss if we didn’t use tools like this to sanity-check the mechanics of our interfaces.

It’s Not All Bad

As a human interface engineer, I’m paid to look for the 5 or 10 percent (OK, maybe 15 percent) of a program’s interface that just doesn’t work. If you took me too seriously, you’d think there isn’t a good interface element out there—which we all know just ain’t true. (If you took the evening news too seriously, you’d think there’s nothing but famine and pestilence throughout the land.) But it’s catching that 5 to 10 percent—and fixing it—that makes for truly great interfaces

and a truly great user experience. The designers of the applications I've just mentioned did an exemplary job of that, and you can, too.

Till next time,

Doc

AppleLink: THE.DOKTOR

Pete Bickford, an engineer in Apple Business Systems human interface group, will no doubt return to his grumpy ways in next month's column. He and the editors of Apple Directions want you to know that the nice things he has to say this month are strictly his own views and do not represent an endorsement by Apple Computer, Inc.

Configuration Data, Part Two: Monitor Size Matters

This month's Market Research Monthly again presents exclusive information from the Apple Computer, Inc., Spring 1993 study of how Macintosh users around the world set up their systems. Last month, we looked at color monitors versus monochrome monitors in the Macintosh installed base; this month, we look at monitor size.

What we're really looking at, and what you need to consider as you plan which monitor sizes to design your products for, are pixel dimensions for each monitor size. After all, what your application really cares about is the number of pixels each monitor makes available to it. Apple and other vendors describe their monitors in terms of inches, because that's what customers understand; throughout this article, we refer to inches instead of pixel dimensions, just to make it easier to read. To translate inches to pixels, we've included the chart "Inches to Pixels" in the box below.

They Don't All Fit One Size

It used to be so simple: With the original "all-in-one" Macintosh computers, you wrote your application to work on a single monitor, the built-in 9-inch display.

Today, you have to consider no fewer than seven monitor sizes when you plan a new product or update an existing one. It's up to you to ensure that your product looks good when run on any of these monitor sizes. Within your program, you can write code that will tell your application the size, in pixels, of the screen.

Many programs—word processors and spreadsheets, for example—have a relatively easy task; all they have to do is open a document window that fills most of the screen. However, you have to do a bit more thinking if your application opens more than one window (some of which may be floating windows) or if you have a fixed area of content that can't easily be resized (which is the case for many games). In such cases, you may have to draw your windows differently based on the screen size.

Games and educational software developers may also want to note that products written for small screens are often run on systems with monitors larger than the ones they were designed for. In such cases, it's considered proper Macintosh design etiquette to include code in your product that blacks out the part of the larger screen not being used by the application window.

To help decide how to lay out your display, you need to know how many customers in the target market for your product use each monitor. Does the target market clearly prefer one size monitor over the others? If so, you might consider optimizing your product for only that monitor size. If not, you might have to spend additional time and money being sure your application looks good on monitors of different sizes.

The Numbers

"Monitor Size By Market," on this page, shows Apple's own data on monitor size within each market, worldwide. As you'll see, the original 9-inch monitor built in to original "classic-style" computers—the Macintosh Plus, SE, SE/30, and Classic® models—is still prevalent in every segment of the Macintosh installed base. The 14-inch, 640-by-480-pixel display (which used to be labeled at 13 inches, for reasons we won't go into here) is the other monitor of choice in every segment but education; there, the 12-inch monitor is also a popular choice. Larger, 19-inch and 21-inch monitors are important for users in small and large businesses.

If you design products for desktop users, you'll want to note that the only classic-style computer still manufactured by Apple is the Color Classic II; also, the installed base is increasingly replacing the original systems with 9-inch monitors in favor of models that use larger screens (although the older systems often remain in use). Therefore, in your future plans for products aimed exclusively at desktop users, you'll want to weigh the 12-inch and 14-inch monitors more heavily in your product planning and design.

When you design products for mobile PowerBook users, you'll need to keep in mind that PowerBook and PowerBook Duo computers employ two different pixel-size screens, 640 by 480 and 640 by 400, even though the screens have the same dimensions in inches (that is, 9 inches).

Next month, the third and final part of our configuration study looks at storage capacity and RAM in the worldwide Macintosh installed base. ®

The graphics in last month's Market Research Monthly incorrectly identified a market segment as "Large/medium government." The segment should have been labeled "Large/medium business and government."

Inches to Pixels

| Inches | Pixels | Type |
|--------|--------|------|
|--------|--------|------|

| | | |
|----|--------------------------------|---|
| 9 | 640 by 480 or 640 by 400 | PowerBook and Duo |
| 9 | 512 by 342 | Macintosh Classic |
| 12 | 640 by 480 | For current Macintosh 12-inch Monochrome Display |
| | 513 by 384 | For discontinued 12-inch Color Display for Macintosh LC computers |
| 14 | 640 by 480 | |
| 15 | 640 by 870 | Portrait monitor |
| 16 | 832 by 624 | |
| 19 | 1,152 by 870 | |
| 21 | 1,152 by 870 | |

New Book Describes Practical Solutions to Your Marketing Challenges

Would you like to save money on your marketing programs, reduce the amount of time you spend finding just the right marketing approach, and circumvent a variety of potential marketing pitfalls? If so, a new book may be just the ticket.

The editors of *Apple Directions*, along with Addison-Wesley, have published *The High-Tech Marketing Companion: Expert Advice on Marketing to Macintosh and Other PC Users* to offer you practical advice for making the most of your marketing resources. It describes concrete, practical, and productive solutions to the marketing hot potatoes described in other marketing books: tried and true, try-this-then-do-that, and this-works-but-that-doesn't everyday-life techniques.

The book's chapters are culled from the Developer Outlook and Marketing Feature columns published in *Apple Directions*; each chapter is a self-contained snapshot of a particular marketing challenge, offering advice that you can put to work immediately. (For a sampling of what the book contains, see the text box "Hot Techniques for High-Tech Marketers" below.)

This compendium of marketing how-to advice was published with the readers of *Apple Directions* in mind; it focuses on solving business and marketing problems and successfully grappling with issues that are most important to you. *The High-Tech Marketing Companion* is the only place you can find this how-to marketing information organized and published in one place.

The expertise in this book comes from many sources throughout the industry. Some chapters were written by recognized industry experts, people who make a living helping developers optimize their marketing efforts and solve marketing problems. Other chapters were authored by developers who face the same kinds of challenges you do and have discovered for themselves what works—and what doesn't. These chapters are mini-case studies that describe how developers have met challenges, grappled with issues, or devised ways to beat the odds of being a smaller company operating in a ferociously competitive

marketplace. Their candid war stories, hindsight, and advice will help make the difference for you, and—who knows?—may hold some surprises in store.

The High-Tech Marketing Companion, by Dee Kiamy and the editors of *Apple Directions*, is published by Addison-Wesley and is available from your local bookseller. Details about mail order will be available in next month's issue of *Apple Directions*.

Hot Techniques for High-Tech Marketers

What are today's marketing hot buttons for *Apple Directions* readers? *The High-Tech Marketing Companion* describes proven techniques for dealing with some of the most pressing issues facing developers today. Here's just a sampling of what you can expect to read about in this new book from Dee Kiamy and the editors of *Apple Directions*:

- how to avoid the ten most common product launch mistakes
- a step-by-step process for choosing and sticking with the best target market
- proven techniques for improving the number and kind of product reviews you get; how to deal with unfair reviews
- a systematic plan for choosing the right name for your product
- developing packaging that helps your product stand out when it's on the dealer's shelf
- increasing the impact of direct-response ads
- creating demos that sell—when you can't be there
- techniques for getting and holding the attention of a national distributor
- how to reap the most benefit from market research done on a limited budget
- a blueprint for building an effective customer support organization
- developing the knack of getting repeat business
- how to increase the effectiveness of upgrade campaigns
- successfully bundling products
- how to get users to work for you—almost for free

- proven techniques for breaking into international markets
- global product launch strategies

Your Ten Biggest Competitors, Part One

To Beat Them, You Have to Know Them

By Regis McKenna

Ask marketing managers to name their primary competitors and they'll rattle off the names of a few other companies in the industry. Marketing managers in the personal computer industry worry about competition from IBM, Apple, and Microsoft. Those in the semiconductor industry worry about Nippon Electric (NEC), Fujitsu, and Texas Instruments.

These worries are, to a large extent, misplaced. Certainly, all those mentioned are tough competitors. But they aren't the toughest competition. They aren't the *real* competition.

The real competition comes from what I call "intangible competitors." These competitors involve ways of thinking and ways of looking at the world. They are obstacles that get in the way of success. When marketing managers resist change, they are facing an intangible competitor. When entrepreneurs begin thinking in the bureaucratic style of "large-corporation man," they are up against an intangible competitor.

Not handling these intangible competitors skillfully is the primary reason marketing fails. If companies can deal with these competitors, they are better equipped to succeed. I have identified ten intangible competitors that all companies confront, regardless of what industry they are in. They 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

Competitor 1: Change

Our society is in a perpetual state of change. Everything is changing. Companies change. One day the newspapers carry a story about a computer company hitting \$100 million in sales. A few weeks later they carry a story about the same company going bankrupt.

Industries change. Deregulation has had a major impact on the structure of the telecommunications and airlines industries. It will completely alter the financial and banking industries in the next decade. Global competition has changed the way companies do day-to-day business, compete, and get financing. One decade ago the names Microsoft, IBM, Compaq, and Sun had no meaning in the small-computer business. Today they are the major influences in the industry's technology and markets. The software industry has undergone an even bigger change. Ten years ago the industry included a few hundred companies. Today there are thousands.

Products change. Today it seems that every product is becoming "smart." Microwave ovens have microprocessors in them. Telephones have microprocessors in them. Even toys have microprocessors in them. With these microprocessors tucked inside, familiar products take on new traits and perform new tasks. Computers themselves are changing too. Today we have computers in all shapes and sizes—personal computers, hand-held computers, portable computers. A decade ago there were few more than 100,000 computers installed around the world. Today 50,000 computers are bought every day.

Distribution channels change. A decade ago no one believed that a computer could be sold through a retail store. Today retail stores sell hundreds of thousands of computers every month. Certainly no one would have believed that a large business could be built based on selling computers by direct response. Dell Computer showed it could be done very well. The value-added reseller and the systems-integration business have also become channels not foreseen a decade ago.

Issues change. Industry issues keep changing as new technologies transform the way we approach our work. Ethics has become a concern of many companies, particularly those in the biotechnology industry. Ethicists, whose main source of employment has been teaching at universities, are now in demand throughout the business world. The shift from producer-driven markets to consumer-driven markets has occurred rapidly in the past decade, creating a host of new issues.

The idea of “customer satisfaction,” championed in the 1980s by people such as Tom Peters and J.D. Power, has become a major issue for corporations trying to figure out what it means and how to implement suitable programs. In the computer industry, the CISC battles of the 80s are giving way to the RISC battles of the 90s. (With the change to RISC, it is expected that the industry will see new performance levels in computing, new software, and new players in the industry.)

These changes constitute a major competitive force. They have a deep influence on the growth and direction of every company. Companies blind to change are doomed to failure. Not keeping pace with change can topple even dominant companies. Companies simply can't afford to stay in the same place.

Business history is full of examples of companies that didn't recognize change in the market, and paid a heavy price as a result. For years, the U.S. auto companies ignored the growing demand for small cars. Japanese companies were attuned to the changing market though, and they quickly stole market share from their American rivals.

The story is similar in the semiconductor industry. In the early 1960s, Fairchild, Philco and General Electric were dominant forces in the industry. None of them recognized the growing importance of integrated circuits, however, and not one of them is a major factor in the industry today.

And the process goes on. Ten years ago semiconductor companies felt pretty secure. They believed the capital intensity of semiconductor manufacturing would prevent new companies from entering the business. But the technology

changes allowed the unbundling of the design of the chips from the processing. This change in the technology created a whole new cadre of “fab-less” semiconductor companies (companies that design chips and have the processing done at silicon foundries). Those companies that didn’t acknowledge this change have been left behind.

The computer industry offers another example. All the major computer companies ignored personal computers in the 1970s. Small start-ups began selling personal computers in 1976, but big companies such as Digital Equipment simply didn’t anticipate a market change. But this market revolution took place right under industry giants’ noses. Many large computer companies, including Burroughs, Honeywell, Wang, Data General, Hewlett-Packard, and NCR, were put in a position of following rather than setting the standards of the new industry. Several have never fully recovered.

In his autobiography, *My Life and Work*, Henry Ford said, “I saw great businesses become but a ghost of a name because someone thought they could be managed just as they were always managed, and though the management may have been most excellent in its day, its excellence consisted in its alertness to its day, and not its slavish following of its yesterdays.” Change has become a part of our lives, with one thing inexorably replacing another. We destroy the old and create the new. In all industries, change is a tough competitor. What can managers—all managers, those in sales, manufacturing, engineering, and marketing—do to cope with this competitor? Two things.

First, managers must constantly question their assumptions. They must ask questions such as: “What am I assuming about the market?” “What am I assuming about the competition?” “What things must happen to make my assumptions valid?” “Under what conditions are my assumptions no longer valid?”

Second, managers must keep their ears to the ground. They must sense change as it is occurring. They must monitor the market, live with it, work with it. Oftentimes changes don’t show up in the numbers and statistics until it is too late to respond to them effectively. Marketeers must develop an intuitive sense of the market. They must work with customers and listen to them. They must meet with

dealers and listen to them. And they must really listen. That is the only way they will spot changes in time to adjust.

Competitor 2: Resistance to Change

Sometimes companies recognize that change is occurring in the marketplace, but they still don't react. For these companies the competitor is resistance to change. Resisting change can be just as damaging as being oblivious to change. In either case, the company can get left in the technological dust.

Examples of resistance to change abound. Consider the case of Gary Boone. In 1972, as a young engineer at Texas Instruments, Boone came up with the idea for a full computer on a chip, later to be called the microprocessor. Boone got a patent on his invention, but he had trouble getting his colleagues interested in his work. He went around TI trying to sell the concept, but he was shot down everywhere. Other people looked at him as a young guy with a crazy idea.

Finally, Boone made enough noise to get a meeting with TI's top guru on computers. Boone went into the office, sat in front of the expert, and explained his idea for a computer on a chip. The expert looked at him with a condescending smile. "Young man," he said, "don't you realize that computers are getting bigger, not smaller?"

There are similar stories involving personal computers. Steve Jobs and Steve Wozniak tried to sell the idea of personal computers to their bosses at Atari and Hewlett-Packard. But their bosses weren't interested. So Jobs and Wozniak started Apple Computer. Intel also had a chance to get in on personal computers early. Several Intel marketing pros went to visit one of the early designers of personal computers sometime in the mid-1970s. They came back and reported: "Bunch of hobbyists. It will never be much of a market."

Such resistance to change can destroy companies. Take a look at the American industries involving the following product lines: autos, steel, consumer electronics, calculators, machine tools, and textiles. In the mid-1960s imports accounted for less than 10 percent of sales in the U.S. market in each of these industries. But American companies in these industries became resistant to

change, while their foreign competitors did not. The result: In 1981 the United States imported 26% of its cars, 17% of its steel, 60% of its consumer electronics (television, stereos, videocassette recorders), 41% of its calculators, 53% of its machine tools, and 35% of its textiles.

What makes companies resistant to change? Sometimes bureaucracy is to blame; sometimes it's just that people are scared and intimidated by new things. People tend to get wedded to ideas. They look toward the past, rather than toward the future. When employees move to a new company or a new project, they bring their histories with them. This experience can be useful, but it also can cause problems. Marketing people often say "This is the way we did it at my old company." This is helpful sometimes, but every once in a while they should say: "Let's experiment and try something new."

The resistance-to-change demon rarely haunts young entrepreneurial companies. Entrepreneurs thrive on innovation and change. They are always willing to experiment with new ideas and new technologies. Resistance to change is anathema to entrepreneurs.

As entrepreneurial companies grow, however, they become more resistant to change. They begin to think more about high-volume production. They develop inflexible systems, processes, and ways of doing business that commit them to doing things in a repetitive, predictable way. That behavior locks them into offering certain products and technologies. They begin to ask questions such as "How do I keep my factories going?" and "How do I keep selling at this rate every month?"

In short, production becomes the central focus of the company. The company begins to worry more about organization and less about serving the needs of the customers. As a result, the company takes on the personality of a large company and becomes less likely to develop innovative new products. Small companies grab the lead in innovation.

The scenario is repeated time and again. The semiconductor memory business provides one example. Intel developed the first semiconductor memory chip and 1K RAM. It clearly established itself as the leader in this new product

category. But when Intel began working on the next generation of memory chips, the 4K RAM, it lost its innovative edge. Intel was committed to the development approach it used with its money-making 1K RAM, but other approaches were better suited to the new generation. A small company called Mostek developed a more innovative 4K RAM, and it emerged as the new leader. And because the two companies did not collaborate on a single standard and work together, the Japanese became the eventual victors.

Mass production inhibits change and innovation. It is based on stability and predictability. When a company moves into mass production it becomes resistant to change. It wants to build up economies of scale. Innovation can disrupt that effort.

Growth companies face a difficult challenge. They must figure out how to move toward high-volume production without losing the innovative spirit that made them successful in the first place. They must continue to see innovation and change as allies, not as competitors.

Corporate politics plays a major role in inhibiting change, particularly in American corporations. Power in the executive ranks seems to be more of a driving force than power in the marketplace. Recently appointed president of Apple Computer, Mike Spindler likes to ask people whether they can name the presidents of several very large Japanese companies. Most people can't. But they do recognize the companies and the industry leadership positions these companies hold. American company managements are more concerned with the appearance of position. Research is done and facts are garnered to support positions rather than to identify weaknesses. Cheerleading is a regular ritual.

Dave Power of J.D. Power told me that Japanese managements don't care for studies that show a simple ranking of car makers based on customer satisfaction. The Japanese managers want to know what they are doing wrong so they can fix things. They want the research to involve asking customers to identify problems, concerns, likes, and dislikes. Things the company can improve on. American car makers like the simple ranking. Even when an American car is far down the customer-satisfaction list, the manufacturer will find a way to brag about it.

Competitor 3: Educated Customers

An uninformed customer is easily satisfied. But there aren't many uninformed customers around these days. Customers today have access to more product information than ever before, and they study it carefully. With technology products, customers are becoming more technology literate.

Customer technology literacy presents a challenge to manufacturers. Customers are no longer pushovers. They want to understand more about the products they buy. They are skeptical and critical, and are often dissatisfied. Manufacturers must meet a higher level of expectations.

Consider the amount of computer information reaching the public these days. A few years ago there were a handful of computer magazines. Now there are hundreds. A few years ago *Time* and *Newsweek* magazines hardly ever wrote about computers. Now they both have computer editors. A few years ago television news never ran stories about computers. But for the introduction of the Macintosh, all three major networks ran stories, as did more than twenty individual stations. Information on high technology has become as integrated into the news as information about the auto industry.

As the quantity of coverage has increased, the quality of coverage has improved. Journalists themselves are becoming more technology literate. Until a short time ago, computer companies could use journalists to spread just about any message they wished. The journalists didn't know enough about technology to critically evaluate computer companies and their products. That has changed. Many journalists use personal computers and are quite knowledgeable about them. When a company introduces a new computer today, journalists want to evaluate the computer themselves. They won't take the company's word about what the machine can and can't do. In effect, the journalist becomes an evaluator for the public.

Hundreds of on-line data bases are available to everyone. We live in the information age and we are often smothered by an excess of information. However, over the next decade information will become increasingly specific and customer programmable. The customer will become even more powerful

because the computer will become more useful as an information-sorting and decision-making tool.

To succeed, companies must turn customers' increasing knowledge of their products from an obstacle into an asset. Rather than battle against a skeptical, critical, and uninformed public, companies should learn from it. They should elicit feedback from customers, then adjust their products and strategies to meet the market needs. An educated customer can only make a business more competitive.

Some consumer-goods companies are already quite successful at using customer dissatisfaction to their advantage. According to a *Wall Street Journal* article, Procter & Gamble phones or visits 1.5 million people each year to ask about P&G products. P&G researchers ask hundreds of detailed questions to find out why customers are dissatisfied and what actions P&G should take to improve its products.

The same article quoted from a study by the U.S. Office for Consumer Affairs: "Many managers view complaints as a nuisance that wastes valuable corporate resources. However, the survey data suggest that complaints may instead be a valuable marketing asset. Responsive companies were rewarded by the greatest degree of brand loyalty."

Technology-based companies should learn a lesson from this. As customers become more knowledgeable—and more critical—about technological products, companies must become more sensitive to customer needs. The philosopher John Stuart Mill once said: "Better to be Aristotle dissatisfied than a fool fully satisfied." Customers of technological products are taking Mill's advice, and companies must adjust.

Competitor 4: The Customer's Mind

People in technology-based businesses tend to think decision making is a simple and rational process. They are wrong. Indeed, when a customer considers buying a product, the decision-making process is neither simple nor rational. All types of fear, doubts, and other psychological factors come into play. Information comes in many disguises.

Winning over the customer's mind is the central challenge of marketing. The customer's mind can be seen either as a competitor or as a competitive tool. Sometimes the customer's mind can act as an obstacle to success. But if companies can understand the customer's mind, they can use psychological factors to their advantage.

All types of things influence the customer's mind. Indeed, the battle for sales is largely a psychological battle. As I explained before, decisions are based largely on intangible factors such as quality, image, support, and leadership. In *Future Shock*, Alvin Toffler describes the psychological battle this way: "For even when they are otherwise identical, there are likely to be marked psychological differences between one product and another. Advertisers strive to stamp each product with its own distinct image. These images are functional. The need is psychological, however, rather than utilitarian in the ordinary sense. Thus, we find that the term 'quality' increasingly refers to the ambiance, the status associations—in effect, the psychological connotations of the product."

Customer attitudes toward a product are not developed by a single event or a single advertisement. Rather customer attitudes develop gradually. They are constantly changing and evolving throughout the decision-making process—and continue to evolve after the decision is made. The "product image" is simply the accumulation of all these attitudes.

The customer's mind can be influenced at every step during the decision-making process. First the people become aware of the existence of the product. They recognize the need for the product. At that point they will try to find out more about the product. They might talk to people who already have used the product, or read reviews written by experts. They might use the product on a trial basis. At each stage their attitudes are modified and reformed. After the purchase, customers' attitudes continue to evolve as they use the product. Customers expect a certain level of product support and product performance. If support and performance fall short of expectations, customer attitudes toward the product and the company will turn negative.

Throughout the entire process, “psychological bogeymen” affect the customer’s mind. These bogeymen include all types of doubts and fears that surround the product, making the customer uneasy about making the purchase. Customers might worry about such things as:

- Is the company going to be around for a long time?
- Am I going to be able to get product support after the purchase?
- Will the manufacturer be able to supply future generations of products?
- Will I be technologically behind if I buy this company’s products now rather than wait for its competitor’s upcoming product?

In winning the battle for the customer’s mind, companies must fight against these psychological bogeymen. They must provide comfort factors that put the customer’s mind at ease. For marketing complex technical products, these comfort factors are particularly important. A company must convince customers that it is financially and technically strong enough to meet all of the customers’ future needs.

At the same time, companies can try to influence customer attitudes toward competitors’ products. With its FUD strategy, IBM works both strategies. *[Editor’s note: FUD is the acronym for Fear, Uncertainty, and Doubt.]* It surrounds its own products with comfort factors and introduces psychological bogeymen to its competitors’ products. Clearly, the strategy has been quite successful.

To succeed in a market, companies also must work to understand the customer’s mind. It is not enough to know what competitive products are on the market and who is using them. Marketeers must also understand the psychological bogeymen and comfort factors that influence the customer’s mind, then use these psychological factors to their advantage.

Competitor 5: The Commodity Mentality

What is good for manufacturing is not always good for marketing. For efficient, low-cost manufacturing nothing beats commodities. By churning out the same commodity product time after time, manufacturers can work all of the kinks out of the production process. As volume increases, manufacturers move down the so-called learning curve, and their costs drop lower and lower.

But a marketing strategy that depends on a commodity mentality can be deadly. Customers usually prefer custom-made, “just-for-me” products. They want their needs satisfied exactly. We are in an age of diversity, and people want to feel as if they are getting something special.

Companies that view their products as commodities will have an increasingly difficult time competing, especially in evolving markets. Companies that sell commodity products can attract customers only by keeping prices low. Competition generally degenerates into a struggle for price leadership, and no one ends up making much money.

How can companies get out of this commodity trap? Meshing the differing needs of manufacturing and marketing isn't always easy, but it can be done. The trick is to view products as more than physical entities. Even if a company manufactures commodity-like products, it can differentiate the products through the service and support it offers, or by target marketing. It can leave its commodity mentality in the factory, and bring a mentality of diversity to the marketplace.

To move away from the commodity mentality, companies must view their products as problem solvers, and then sell the products on that basis. Service adds another dimension that provides commodity businesses with a differentiation. Dell Computer sells IBM clone computers. But by also offering a 24-hour 800 number for support, service guarantees, information via fax, and other services, it has achieved a distinguished position in a commodity business.

In his *Harvard Business Review* article “Marketing Success Through the Differentiation of Anything,” Theodore Levitt describes the approach this way: To the potential buyer, a product is a complex cluster of value satisfactions. The generic thing is not itself the product.... A customer attaches value to a product in proportion to its perceived ability to help solve his problem or meet his needs.

An automobile, for example, is not just four wheels and an engine. It is a product that fulfills customer needs, psychological and otherwise. Manufacturers can differentiate their automobiles according to the needs they fulfill. One can be

positioned as a status product, another as a performance product, even if the products themselves are quite similar. If automobiles were marketed solely on the basis of their specifications (the number of cylinders, the size of the engine, and others), customers would perceive them all as being very much alike. Indeed, specsmanship marketing is a sure sign of a commodity mentality.

The personal computer provides another example. Everyone views the personal computer in a different light. Many managers see it as a productivity tool that provides increased freedom to information users. Some MIS managers see the personal computer as a device that causes information and other resources to be used inefficiently within large organizations. The product is the same, but the perceptions of it are very different.

The perception of personal computers also changes with time. At first the Apple II was seen as a hobbyist computer. Then as a small-business computer. Then as a vertical-market computer—a computer able to serve many different, specialized applications. But the Apple II itself remained largely the same. However, the marketplace has changed, and so has Apple's marketing strategy. Apple has manufactured the Apple II like a commodity. But in its marketing, Apple made the Apple II special to every customer. It stayed away from a commodity mentality.

[This month you've read about the first five of your ten biggest competitors. In next month's issue, we'll complete this discussion with a close look at the final five competitors: The Bigness Mentality, Broken Chains, The Product Concept, Things That Go Bump in the Night, and Yourself.]

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.

AppleDirections

Volume 2, Number 2

Apple Directions, Apple's monthly developer newsletter, 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

Meredith Best, Juan Bettaglio, Pete Bickford,
Pat Calderhead, Alex Doshier, Tim Enwall,
Amanda Hixson, 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, A/UX, LaserWriter, MacApp, Macintosh, and MPW are trademarks of Apple Computer, Inc., registered in the U.S. and other countries. AppleSearch, DocViewer, Macintosh Quadra, MessagePad, Newton, OpenDoc, PowerBook, PowerBook Duo, QuickDraw, QuickTime, System 7, 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. Claris, MacPaint, and MacWrite are registered trademarks of Claris Corporation. Classic is a registered trademark licensed to Apple Computer, Inc. DEC is a trademark of Digital Equipment Corporation. PowerPC is a trademark of IBM Corp. UNIX is a registered trademark of UNIX System Laboratories, Inc. All other trademarks are the property of their respective owners.

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