The Developer Business Report *September 1993*

AppleDirections

Inside This Issue

Apple News

Special Report: Apple Ships Newton MessagePad

"Newton is here!"

So said Apple Computer, Inc., Chairman John Sculley at 10:30 A.M., August 2 at Boston's Symphony Hall, where he announced "a revolution in communications" while showing off the first Newton personal digital assistant (PDA), the MessagePad communications assistant.

The MessagePad began selling that day—by the thousands—at the Macworld Expo in Boston as well as at Apple resellers in Boston and New York. By the time you read this, the MessagePad will be on its way to more than 2000 computer retailers and consumer electronics stores throughout the United States. Apple has not set a retail price, but it's expected that resellers will price the MessagePad from $699 to $949.

Delivery of the 7.25" x 4.5" x 0.75", 0.9-pound hand-held communications assistant signaled a radical turn in Apple's business, and it could signal a turn in yours, as well. Apple will be selling the MessagePad and, soon, an entire line of PDAs into a new, large market, a market that will be open to your software and hardware products that

Please turn to page 6

Strategy Mosaic

Why Newton Will Succeed

And How You Can Get Started

By Gregg Williams, Apple Directions *Staff*

The first Newton PDA (personal digital assistant), the Newton MessagePad, is finally out, and I'm pretty excited by the business and development possibilities. I sat in on the first Newton programmers' kitchen, and I'm impressed by the state of the development tools, not to mention the MessagePad itself.

The news story immediately to your left raises the curtain on the Newton MessagePad product. My purpose here is to tell you the strategy story behind the Newton business, a story that I hope will persuade you that this is a business you want to be part of.

It's a big enough story that I'm going to take two months to tell it. This month, I want to convince you that there is a *large* Newton market out there, say a bit about the first Newton customers, and talk about how Newton developers and Apple will interact. If you're even thinking about getting into Newton development, you need to read this column—carefully—before creating your business plan.

Next month, I'll give more detail about why Newton software is fundamentally different from personal-computer software, how to leverage your skills to make sound Newton products, and where Apple thinks the main

Please turn to page 4

AppleDirections

Volume 1, Number 4

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
Juan Bettaglio, Cassi Carpenter, Alex Dosher, Stacy Moore, Rilla Reynolds, Caroline Rose, Adam Samuels, Tracy Swedlow, Anne Szabla, Wendy Tajima, Jessa Vartanian

Manager, Developer Press
Dennis Matthews

Manager, Developer Support Information
Greg Joswiak

Production Manager
Diane Wilcox

PrePress/Film
Aptos Post

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

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

Editor's Note

A Parable

I'd like to tell you a story.

When I was about ten, my little brother, Andy, had a teddy bear named Newton (really). Even though he was small—no taller than ten inches, if I remember correctly—Newton was the greatest. Hand-made by a famous German maker of stuffed animals, his arms and legs moved, his fur seemed real, and his face was just on the verge of saying something.

The maker's handiwork wasn't enough for my brother, though. Andy liked Newton so much that he brought him to life. With the help of my grandmother, he knit Newton a white sweater and sewed him a pair of pants out of red velvet. He gave him a home next to his bed, and Newton was always at the center of a coterie of stuffed friends my brother collected.

I didn't have half the imagination of my brother; to me, stuffed animals were stuffed animals. But I was so taken by Andy's vision that I began to see Newton as real. At night, when we were supposed to be falling asleep, we'd have parties with our stuffed animals. We made a stage ringed with tiny Christmas lights for Newton and his gang so they could put on plays for our entertainment.

Newton was great all by himself, but what truly made him come to life wasn't the German toymaker but my brother's imagination and the clothes, the stage, and the other flights of fancy. Newton still sits on a shelf in Andy's home, although it's been years since his last performance.

You've probably guessed the point of my little story by now; Apple, of course, has just released its own Newton, the MessagePad, and we think it's pretty terrific, all by itself. But like the bear, to bring the vision for Newton fully to life, we and Newton's customers will be counting on your imaginations. We want you to have the same "flights of fancy" that made the Macintosh computer so successful and to help all of us figure out what Newton technology can really do.

In this month's issue, we focus on telling you what the best minds at Apple and some other places have done so far with Newton. The MessagePad, however, is but the first stepping stone on a long path. At its introduction in Boston, Apple Computer, Inc. Chairman John Sculley was already talking about a "wide-ranging family" of Newton devices: MessagePad devices of different sizes, electronic pen pads built into phones, display fax phones, Newton pads in electronic books, and Newton devices in cars to provide maps and guidance systems.

We challenge you to go us all one better. Even though some big names have already announced plans to develop Newton products, now is definitely the time when one person with a great idea can make a huge difference.

Paul Dreyfus
Editor

P.S. Some years after putting Newton the bear on his shelf, my brother became a graphic designer at Apple where, among other accomplishments, he designed the light-bulb logo you'll see on every Newton.

Apple Directions On-Line for October

The October issue of *Apple Directions,* shipped to you in its paper form in mid-September, will be available on AppleLink as follows:

September 1—Preliminary draft copy	September 15 —Final copy

To view *Apple Directions* on line, follow the AppleLink path Developer Support:Developer Services:Periodicals:Apple Directions:Apple Directions October.

IndustryWatch: News & Perspective

If We Build It, Will They Come?

By Amanda Hixson, Consultant, Instant Insight

Bouncing Back

A few years ago many folks were ready to write Compaq Computer off as a casualty of the personal computer price wars. After many years of record-setting growth and earnings, the company's earnings plummeted faster than a bungee jumper from a bridge. For awhile there, it looked like the company was making a 100-foot jump using a 110-foot bungee cord. Then—

WHOOSH!

Last month the company snapped back quicker'n spit on a griddle. It announced earnings of $102.3 million for the past quarter, up from $29 million for the same period last year. Quarterly sales nearly doubled, shooting up to $1.63 billion from an anemic $826 million a year ago. And, while some computer companies struggle with layoffs and cutbacks, Compaq is said to be gearing up for 150 to 200 new jobs.

What happened?

You can spend a lot of time reading annual reports, analyzing balance sheets and profit-and-loss statements, examining management changes, and a whole bunch of other things looking for the answer to that question. Personally, I think the reason for Compaq's amazing comeback is that they paid attention to a fairly simple theorem: The company figured out what business it was in and then determined how to succeed in that business. Arithmetically speaking:

Well-Defined Business Purpose
+ Correct Implementation Model
= Success.

Now, if a few other large personal computer manufacturers follow the same simple theorem I've postulated for Compaq, they may find that current painful experiences such as layoffs and shrinking margins are correctable. In other words: It can be done.

Two Mints in One?

Of course, a theorem is only an idea that is demonstrably true or assumed to be so. Therefore one needs to find a proof for the theorem in order to validate it. Finding a proof often takes a great deal of testing and analysis coupled with a growing number of cases that work within the bounds of the theorem.

Fortunately for me, there are always folks willing to challenge a theorem. Take Louis V. Gerstner Jr., IBM's new chairman and chief executive officer. After three months of getting-to-know-you leadership, Louis V. decided to slam the brakes on IBM's very large decentralization effort, which was in motion when he joined the company.

According to the *Wall Street Journal,* Gerstner wants to have his cake and eat it too. Instead of completing the subdivision of IBM into smaller, autonomous divisional units responsible for justifying their continued existence based on their respective abilities to generate profits, Mr. Gerstner apparently wants to retrogress to the old solar-system approach with eight product divisions orbiting the IBM sun.

The *Journal* says Gerstner's thinking is that you can be too centralized or too decentralized, and that what I call "sub-centralization" is possible through an entrepreneurial divisional approach: Act like an independent, startup business within the gravitational pull of the larger corporate body. This back-to-the-future approach is justified in the *Journal* article by comments such as "a lot of what went wrong [previously] was execution, not strategy," made by Gerald Czarnecki, Gerstner's new chief of staff.

To me, an entrepreneurial division is an oxymoron: The word *entrepreneurial* implies a risk/reward structure not available within the confines of the corporate/division relationship. If I'm correct, this validates the Compaq theorem proposed earlier. IBM is simply not letting its divisions prove their ability to independently define their business purpose, implement strategies, and succeed or fail on their own.

Field of Screams

As a consultant, I specialize in project and process management. The concepts of project management and process implementation often seem alien, if not downright scary, for many people. I won't bore you with reasons why you, and your technical staff, shouldn't be afraid of correctly implemented processes. I will, however, point out that much of project management involves simple common sense.

That's why it bothers me so much to see stuff in the press that resembles the following paragraphs. I swear I'm not making this up; although I have doctored the quotes, basing them on the content of an original article and eliminating any real names in order to save the involved people embarrassment, this is based on a real article I recently read.

Despite schedule adjustments and rumors of the project's death, Mr. Big Guy, head of Monkeyboy, Inc., said: "We are going ahead with FarFinkle. This baby will be on the market before the end of the year. I must point out, however, that it is still a substantial risk for us, but FarFinkle could, over a few years, be as large as the entire Monkeyboy Company is today. Then again, it might hit the market and be dropped like a hot rock if it's not what people want. When you get right down to it, customers express their feelings by the number of bucks they'll give you from their wallets for whatever gizmo you're trying to sell them."

In fact, Monkeyboy is still struggling with a couple of issues around FarFinkle. "With a nearly completed product, we need to ask ourselves two questions," said Mr. Big Guy. "What's our marketing plan? and who is the customer?"

I know, you probably still can't believe that any company would build a product this way and expect to be successful. But, whether you believe it or not, more projects than I care to think about are managed using the "build it and he will come" philosophy. In cases like this, however, "he" turns out to be a Jackson, Penniless Moe,

IndustryWatch

instead of Shoeless Joe, and the field is a nightmare, not a dream.

What makes the above example all the more painful for someone in my line of work is that the individual represented in my dramatization as Mr. Big Guy later goes on to say something like this: "We used to just build products and then worry about figuring out who the customer was. Now we think about who the customer might be and then build a product."

The preceding quote sounds much better than the earlier ones, but both examples contain a fundamental flaw: They assume there is a customer for whatever it is *you* want to build and ignore the possibility of building something *your customers* might want you to build.

Something that, oh my gosh, you might not have thought about, or something that might even be a logical extension of one of your existing products. An extension you might not see because you're too close to the product.

The lesson: Spend time *up front* making sure your product concept is valid, ensuring that there really is a market for it, and designing the product to be useful to the target customers. (You probably have many different customers for every product you build, some of whom you're not reaching because you didn't include their needs in your product specification.) For every day of this advance planning, you'll probably save several days on the back end that you currently dedicate to rework, bugs, and design changes.

Don't do it for me; do it for your customers and your own bottom line. ♣

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).

Strategy Mosaic

Newton Will Succeed

continued from page 1

market opportunities will be. You'll want to read next month's column before you design your first Newton software.

(To prepare for these columns, I talked to several marketing and technical people at Apple, and I audited a five-day class on programming Newton. Among the people I talked to are Philip Ivanier, manager of development relations, and Ken Wirt, director of marketing for Apple's Personal Interactive Electronics Division. Most of the information that follows comes from Philip and Ken.)

The Newton Customer

First, a real-life parable: Two companies—a computer manufacturer and a chain of movie-rental stores—jointly announced their new technology for distributing movies electronically. The day after, several major movie companies said that they strongly opposed this new technology. It seems that the first two companies had never even consulted the other companies before making their announcement.

To devise Newton, we took a lot of time to listen to customers before deciding what kind of product to create. Before developing Newton, we studied people at work, at home, and at play, with the following question in mind: "What kind of people are most likely to benefit from having a PDA?"

We found that the potential Newton customers share several important traits:

- They want to have a PDA with them all the time.
- They are always moving from place to place.
- They need to keep track of many small, miscellaneous pieces of data.

Apple also did some market research that involved one-on-one interactions with over 500 people in the United States and Great Britain, including Macintosh users, DOS/Windows users, and people who didn't use personal computers. (This was a "blind" survey, so the interviewees did not know what company was asking their opinions.) During the interview, these people got to use a prototype Newton device.

The research turned up some truly unexpected things about the people who expressed the greatest interest in the prototype unit they were shown:

- Both women and men were equally likely to express an interest in buying and using Newton.
- The majority were not traditional early PC adopters. (This means that the potential audience for Newton products is bigger than the relatively small number of people who always buy new technological toys.)
- The majority were not Macintosh users.

These are all good signs! Men, early adopters, and Macintosh users are all good markets, but you have already reached them and sold to them. The above findings indicate that the market for Newton devices is both outside the current personal computer market and, more important, *much, much* larger than the current market.

We also received gratifying feedback from the interviewees: "It's like nothing I've ever seen before." "This is the wave of the future." "It'll sure make my life easier." And my favorite, "I don't care what they cost—I want two of them." (Guess what? A woman said that after deciding that she couldn't share a single Newton with her husband—he'd need his own.)

The Impetus Toward Newton

In business, market forces are forcing businesses and corporations to do business better with fewer resources and employees. (The nineties buzzword for this is "re-engineering.") We've found great interest for Newton devices in several markets, especially health care, pharmaceuticals, and financial services. Employees who take a Newton device with them can be more efficient, and they can collect needed data with fewer errors. (There are no transcription errors because the data can go straight from the Newton device into a desktop computer.)

Newton devices make something new possible—what I call "anecdotal data." Once you put the MessagePad in the hands of your employees (or even your customers), you can arrange to have it return all sorts of data that would be impractical or too expensive for you to go out and gather in other ways. So not only does the Newton device help you (and others) get the job done, it also completes the loop and delivers valuable information about your customers.

A Newton device can often replace another piece of equipment and do its job better. For example, imagine a Newton device connecting to a pager or a hand-held dictating machine. (The Newton MessagePad has a slot for business card–sized PCMCIA devices.) In both cases, the Newton device can add both "intelligence" and a much improved user interface.

On the personal level, people want Newton devices to make their lives a little easier. Women often picked up on this: They indicated that they often have to coordinate their activities with those of a significant other and kids. And doesn't that match the "all-the-time/mobile/miscellaneous-data" profile mentioned earlier?

New Distribution Channels

No doubt about it—Newton is a new kind of product, you will write a new kind of software for it . . . and, depending on your situation, it may change the kind of working relationship you have with Apple. (Apple's official way of saying this has been that Newton represents "a different business proposition" from the Macintosh one that everybody currently knows.)

Electronic distribution will become a legitimate channel for commercial Newton software, and Apple will ensure the existence of at least one such channel by operating one itself. Apple Online Systems, a division of Apple, will establish a new service that will allow any Newton user with a modem and a credit card to buy Newton software by downloading the software into the Newton device and paying for it in one automated phone transaction.

Electronic distribution is a new way of getting software to customers, but it's just one component of Apple's entire strategy. Apple is also working on a new model of software distribution that a lot of retailers—including department, buying-club, consumer-electronics, and office-supply stores—are interested in. Apple eventually expects to recruit book and record stores into selling Newton titles.

Apple will be distributing to retail outlets both Apple-label and third-party Newton titles (more on this later) and other products from Sacramento, California. Stores will be able to demo a large number of Newton titles and will keep at least one copy of each title on hand. When they run out of one or more titles, stores will be able to order them from Apple and, in the United States, be able to replenish their stock in one day.

By minimizing the space and money usually tied up in maintaining product inventory, Apple has come up with a distribution method that encourages retailers to carry Newton titles and, equally important, to carry a larger number of titles. In this and other ways, Apple is working to make sure that if you create a Newton title, Newton users will know about it and be able to obtain it easily.

Selling Newton Products

Because you will have multiple ways of getting your Newton products to customers, you should choose the one that fits your company size, business objectives, and resources. Apple is trying to "level the playing field" so that a small company (down to that extremely important "one person with a good idea") will be able to start a sustainable business. In fact, several of the companies introducing the first Newton software products are small, startup firms.

If you have the resources and are willing to take the risk, you can certainly create and distribute your own Newton titles. However, depending on the details, you may find that having Apple distribute your software makes it possible for you to consider developing for Newton. (I don't yet have the details on Apple's distribution plans—*Apple Directions* will let you know as soon as they're available—but I wanted to give you the above information so that you can begin making your plans.)

If you're good (and lucky) enough, there may be a third avenue of sale for your product: Get Apple to buy it from you. Apple will sell a selected line of Newton titles through a new label, Starcore, and Apple will work with selected outside companies to bring them to fruition.

How do you get Apple's attention? Well, if I knew that, I'd be a software agent—the kind that gets 10 percent of your profits—not an Apple employee. But a good first step is to write the Newton group at AppleLink address NEWTON.DEVS and tell them what a great idea you have. For a handful of talented developers, such a message might be the beginning of some help from Apple. (Remember, I said "a handful.")

Another good step is to purchase a beta copy of the Newton Toolkit, the development system you use on a Macintosh computer to create Newton software. The beta version of the toolkit is available now from APDA in advance of the final product release later in the fall.

One important point: The Newton software that Apple sells will be general-purpose software of interest to many people. Apple plans to leave vertical markets to developers, because it makes sense for you to exploit them and Apple not to. We need your expertise and numbers to create the many useful software titles that Apple cannot create. And by enlarging the breadth of Newton software, you help ensure the success of the product line itself.

Getting Software Into the Newton MessagePad

You need to decide how your software will get into Newton devices—after all, they don't exactly come standard with 3.5-inch floppies. Here are the options I know.

First, customers can download software into Newton devices through a modem and phone line.

Second, if your customers have a Macintosh or Windows personal computer, you can give them a floppy disk with your software on it. Every Newton device will contain a serial port and can be connected to a Macintosh or Windows PC using a product called the Newton Connection Kit (to be shipped this fall). Software on the floppy disks can be read by the PC and then loaded into the MessagePad. (Information can also be exchanged between the MessagePad and the PC by using the Newton Connection Kit.)

Third, you can produce PCMCIA Type 2 ROM cards, which can contain several megabytes of programs and data. Users simply swap cards in and out of a convenient slot at the top of the MessagePad.

The Market's There . . .

. . . so *what* do you build for it? Next month, I'll look at why Newton software is different and what to keep in mind when designing your first Newton software. ♣

Strategy Mosaic is a monthly look at pieces of Apple's overall strategy. This column is based on information from managers throughout Apple Computer, Inc. Gregg Williams has been writing about Apple technology since 1982. You can contact him at GREGGW on AppleLink, or by phone at (408) 974-3264 (in the United States).

Apple News

Newton MessagePad

continued from page 1

augment Newton's innovative technology.

At first, that market will be principally located in the United States; if you're outside the United States, you'll be able to develop for the U.S. market, but you'll have to wait until some time next year to see Newton devices localized for other languages. For more about the market, and how you can get started reaching it, see this month's Strategy Mosaic on page 1.

The key word in the last two paragraphs is *open,* because that's what Apple's Newton technology is, a point emphasized during the introduction festivities in Boston. Apple is licensing Newton technology to a variety of major computing, communications, and personal electronics firms so that they can incorporate it into their products.

Newton is also open for development. In addition to the MessagePad, Apple is selling a beta version of the Newton Toolkit, the development environment you use on the Macintosh computer to create Newton software. You can obtain the beta Newton Toolkit from APDA now (see page 36 for APDA ordering information); the final version will be available later in the fall, along with a special support program for Newton developers. You'll be in good company: John Sculley told the Symphony Hall crowd that 1,500 developers have already shown an interest in creating products for Newton.

More in a moment about the firms that have licensed Newton technology and begun work on Newton software; you've probably waited long enough to hear about what the MessagePad is and what it does.

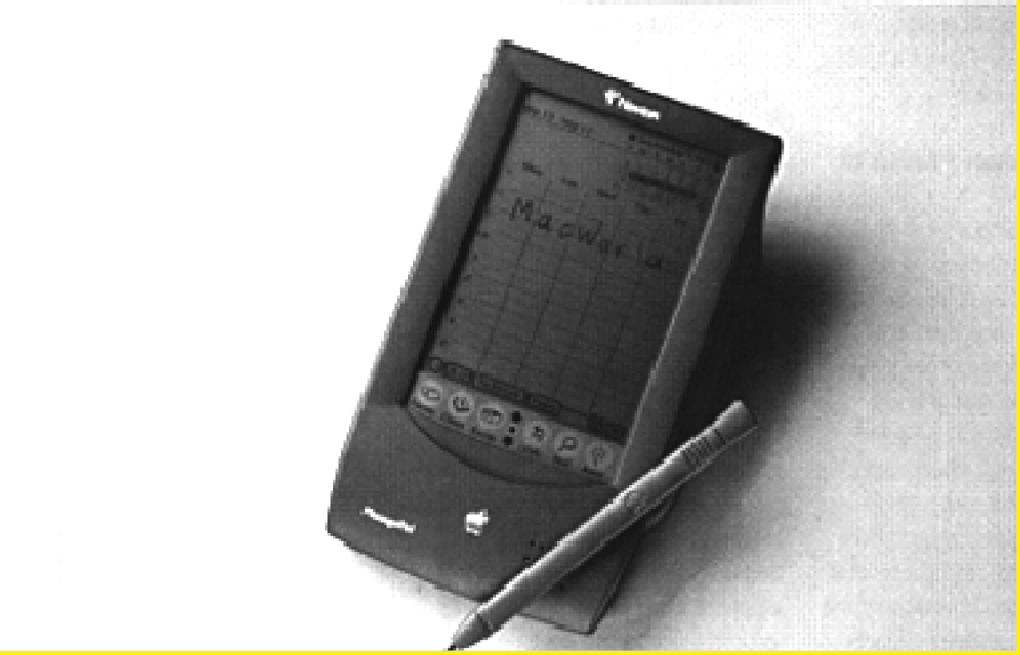
MessagePad Features

As you read this, keep in mind that the MessagePad communications assistant (shown on this page) is only the first product to utilize Newton technology. As John Sculley said in Boston, "The Newton product development pipeline is full."

Although Apple is positioning the MessagePad as a communications assistant, it incorporates the same basic technology expected to be part of most or all Newton PDAs—technology that helps users capture, organize, and communicate every kind of information they typically encounter, either at home, on the move, or in the office. This includes, but certainly isn't limited to, notes to yourself or others, names, addresses, telephone numbers, appointments, ideas, laundry lists, shopping lists, directions, and even rough sketches.

The MessagePad and all other Newton PDAs can also calculate and manipulate information, in ways that will be as clever as those of you who develop applications. In fact, *clever* is a great word to describe Newton; Newton software lead engineer Donna Auguste told the introduction audience, "The MessagePad is easy, fun, and powerful; nothing is comparable."

And batteries are included! You can take it with you wherever you go; product literature says that its four AAA batteries (which can be recharged with an optional pack) will last up to one week, depending on the user. It also ships with an AC adapter. (See "Newton MessagePad Fact Sheet" on page 7 for technical details about the MessagePad.)



Apple's first Newton personal digital assistant: the MessagePad communications assistant.

• *Capturing Information.* Using the MessagePad's flat "passive pen," you write letters and numbers directly on the screen; Newton "recognizes" your handwriting and transforms it into typed text, although you can choose to leave your notes in handwritten form. When you first use the MessagePad, it may have difficulty recognizing your handwriting—as a skeptical engineer asked, "How can a machine recognize my handwriting when even I can't?" However, the longer you use the MessagePad, the more it learns your writing. Also, it includes an application that teaches you how to write more recognizably. (Someone who's used the MessagePad says that it recognizes cursive better than printed writing.)

You can also sketch on the

MessagePad; again, it will either clean up your drawing or save it as is. If you don't like what you've written or drawn, you simply scratch it out with the pen and it disappears. And, when you want to throw a note away after you've created and saved it, the MessagePad shows you an animation of the note being crumpled and thrown in a wastebasket.

• *Organizing Information.* There are two important aspects of how the MessagePad organizes information. First, with technology Apple is calling *Newton Intelligence,* the more you use the MessagePad, the more it learns and mimics how you work and organize. The MessagePad learns your accustomed format for setting up memos, for printing, and for recording dates in your calendar, to name a few often-repeated tasks. As Donna Auguste said during the introduction, "When you use it, Newton becomes a reflection of you."

Also, unlike the Macintosh, the MessagePad does not use a WYSIWYG ("what you see is what you get") scheme. For example, suppose you're keeping track of expenses on a business trip. (This is a hypothetical example, because an application for building an expense report isn't yet available.) You'd use the MessagePad the same way you would use scraps of paper and receipts, writing down who you had lunch with and how much it cost, how much you spent on an audio tape of a conference session, or that you bought toothpaste and aspirin in the hotel's gift shop.

The MessagePad knows to record the bits of information in appropriate places on its expense report form, which is stored in ROM. Later, when generate your report, the information you recorded at random has been transformed and calculated into finished form. The form the MessagePad uses to generate the information is not the same as the form you used to enter it. In this way, it's unlike a WYSIWYG expense application, in which you'd enter the bits of information into specific places in the form on the computer screen. (More about this in next month's Strategy Mosaic.)

To help users stay organized, the MessagePad ships with an assortment of information and productivity tools, including a calendar, an address book, timezone maps, a calculator, and currency exchange information.

• *Communicating Information.* Apple thinks that Newton's communications technology is pretty significant—it's calling the MessagePad a *communications assistant*—and with good reason. John Sculley told the audience attending the introduction that "Newton will be big in the current analog communications world. We're not waiting for the digital highway to be complete."

The MessagePad's Newton Communications Architecture allows users to send, receive, and share information through virtually any currently used communication channel—such as telephone lines, wireless networks, and computer networks. Not all the applications and services that will one day make this possible are in place yet, but once they are, users will be able to easily add them to their MessagePad devices.

For now, the MessagePad can communicate by fax using an optional modem connected to a telephone line. Infrared technology enables the MessagePad to

Newton MessagePad Fact Sheet

Here are selected technical details about the first Newton product, the MessagePad communications assistant.

Newton MessagePad Hardware Architecture

- ARM 610 processor at 20 MHz
- Apple custom system ASIC
- low-power, reflective LCD display (336 x 240 pixels)
- transparent tablet with passive pen
- 4 MB of ROM
- 640K of RAM
- one PCMCIA type 2 card slot
- localTalk-compatible serial port
- low-power, half-duplex, infrared transceiver (9600 baud at 1 meter)

Communications Capabilities

- print to Apple LaserWriter and StyleWriter printers with serial cable and to popular parallel printers using Newton Print Pack
- fax with optional fax modem
- beam by means of infrared technology up to a range of 1 meter
- handle telephony needs with assistance for intelligent auto-dialing
- receive and store wireless messages using optional Messaging Card

Size and Weight

- height: 7.25 in. (184.75 mm)
- width: 4.50 in. (114.3 mm)
- depth: 0.75 in. (19.05 mm)
- weight: 0.9 lb. (0.4 kg)

Power Requirements

- power adapter:
  —Universal, 100 to 240 VAC, 50/60 Hz
  —Four plug configurations: U.S., U.K., Europe, and Australia
- battery type: alkaline or nickel-cadmium
- power source: AC adapter (included) or four AAA batteries (included) and optional Rechargeable Battery Pack

System Requirements

- Works with personal computers running Microsoft Windows, version 3.0 or later
- Works with Macintosh computers running system software version 7.01 or later

Printer Support

- Works with Apple StyleWriter and StyleWriter II, all Apple PostScript laser printers from LaserWriter Plus up, and Apple Personal Laser Writer printer
- Works with most popular PC printers, including Hewlett-Packard LaserJet, Hewlett-Packard DeskJet, Canon BubbleJet, and Epson LQ and FX using the optional Print Pack ♣

beam messages to other Newton devices up to a meter away. Users can also print any document the MessagePad creates on most printers, through a direct connection or over a local area network.

Users will soon be able to connect the MessagePad to either a Macintosh or Windows computer, using the optional Newton Connection package, to transfer, synchronize, back up, and update information between the MessagePad and the computer. A technology Apple calls *Smart Synchronization* automatically updates information between the MessagePad and the computer when they're connected to each other. Other special software shipped with the connection package creates a "virtual Newton" on the computer screen to allow the user to create, view, and edit Newton information on the Macintosh or Windows computer. Newton Connection products will be announced at a later date.

Other communications technologies to be made available to MessagePad users in the near future include electronic mail—through a product called *Newton-Mail* —and the ability to receive fax messages.

Also, MessagePad users in the United States will soon have access to wireless messaging services. Apple recently announced an agreement with BellSouth Corporation and its subsidiary, MobilComm, to provide wireless messaging for Newton products through MobilComm's nationwide paging network. Their plans call for integrating cellular phone technology into Newton; the two companies expect to launch a paging service for the MessagePad later this fall.

Newton Software

That sums up what the MessagePad does (and soon will do) on its own. There's literally no telling what the MessagePad and other Newton devices will one day be able to achieve, once developers get their hands on them and design the software that will truly bring the Newton vision to life. For the short term, we can tell you the capabilities the first software products will bring to MessagePad users, in the hope that your imaginations will be stirred to devise your own innovative titles. The products discussed here have only been announced and are still in development; dozens are expected to ship by the end of the year.

As you'll see, the field for software and MessagePad-based solutions is wide open—to new ideas and new players. John Sculley predicts that Newton will create an "extraordinary business." Interestingly enough, a product initially destined for use by individuals is also finding a potentially very promising place in vast enterprise schemes, as briefly described in the section "Corporate Applications" later in this article.

- *Apple Titles.* First off, Apple has its own six software titles, to be marketed by Starcore, the publishing group recently launched within Apple's Personal Interactive Electronics division. Some of Apple's titles will be delivered on PCMCIA cards that fit into the card slot at the top of the MessagePad; others will be shipped on a floppy disk and can be downloaded from a personal computer using the Newton Connection package. To read more about Apple's MessagePad titles, see the related story "First Newton Titles From Apple's Starcore Publishing Group" on the next page.

- *Third-Party Titles.* In addition, more than 20 developers have announced MessagePad products, ranging from specialized realtor systems to titles for sports enthusiasts. The developers include a variety of companies, from established Macintosh developers such as Claris® and CE Software to startup companies issuing their first products.

Apple Directions saw demos of two of the third-party products, both of which were impressive, even at the beta stage of development. One of them is a calculator that includes conversion tables for practically anything you can imagine; with the software, the MessagePad can instantly figure equivalent kitchen measures, tell you how much of a tip to leave a waiter, and convert a date to a day of the week.

Another product provides intelligent drawing assistance. You can draw a rough shape on the MessagePad; it will then figure the shape's dimensions if you tell it the measurement of one side, change the shape's proportions if you change a side, and calculate the degree of its angles and recalculate all the angles if you change one of them.

- *Corporate Applications.* There's also a third category of applications under development that involve integrating the MessagePad into corporate information systems. Companies such as Coca-Cola, Monsanto, American Express, and the Department of Defense have announced intentions to use Newton technology and the MessagePad to help streamline processes, operate in a world of shortening product life cycles, and become more responsive to customers.

American Express, for example, is developing applications for customers who use Newton devices, while the Department of Defense, together with Apple and KPMG Peat Marwick, is investigating applying the MessagePad to its health-care operations in a project called *ProMED.*

Licensing Newton Technology

As mentioned already, the Newton business model has been open right from the start. Long before announcing the MessagePad, Apple began licensing Newton technology to others so that they could create their own Newton devices.

In other words, Apple will not be the only supplier of Newton PDAs. Instead, Apple will lead the effort, along with some heavy hitters, to set a standard in the hand-held computing industry. A variety of leading corporations will help create the market for your products, since it's anticipated that software, titles, and accessories will be able to be used by their Newton PDAs as well as by Apple's.

Among those who have licensed Newton technology are Sharp and Kyushu Matsushita in Japan and Motorola in the United States. Siemens/ROLM will incorporate Newton technology in its forthcoming NotePhone product.

Special Thanks

Introducing Newton in Boston, John Sculley compared the delivery of the MessagePad to the delivery of the first Macintosh computer nearly ten years ago.

"Ten years ago, we started a revolution for the desktop; today, we're beginning a revolution for the pocket.

"Newton will help the rest of us cope a little better with our ever-changing, ever more complex world," he added. "It's the technology metaphor for today's mobile society."

During his speech, Sculley paused to give special thanks to the key individuals who undertook what he called Apple's "most ambitious development project ever." We would also like to mention these individuals, some of whom are still at Apple, others of whom have departed: Steve Sackoman, Jean-Louis Gassée, Larry Tesler, Steve Capps, Mike Culbert, Michael Tchao, Donna Auguste, and Gaston Bastiaens. Thanks to you, and to the many others who made Newton a reality.

First Newton Titles From Apple's Starcore Publishing Group

Apple Computer, Inc., has announced both a new publishing group and its first six titles, all of them application software for the Newton MessagePad communications assistant. The Starcore publishing group, recently introduced within Apple's Personal Interactive Electronics (PIE) division, worked with a variety of publishers, including Random House, GeoSystems (a subsidiary of RR Donnelley & Sons), Pensee, PenMagic Software Inc., and Blank, Berlyn & Co. to create software for the MessagePad, now beginning to ship throughout the United States.

"This announcement underlines the fact that the Newton business is not just about hardware," said Ken Wirt, director of marketing for the PIE division. "We have been working closely with developers in this venture, and we are extremely pleased with this first portfolio of titles. I believe that Starcore is now well positioned to contribute significantly to the acceptance of the Newton family as the principal standard for the PDA market."

Here are the first titles from Starcore Publishing, which we'll describe here to give you an idea of the current range of Newton capabilities. Apple encourages you to create your own MessagePad applications and significantly broaden this range. We've included suggested retail so you'll know what you can charge for comparable titles. These titles are expected to be available to customers by early fall.

- *Fodors 94 Travel Manager: Top U.S. Cities* includes 500 locations per city, such as restaurants—detailing the type of cuisine, the credit cards they accept, and other travel information and business services—hotels, sights to see, and airport information. Covers eight cities, with detailed interactive maps highlighting destinations and routes. Suggested retail price: $119.95.

- *Fortune 500 Guide to American Business* is an enriched electronic version of the Fortune 500 and Service 500, giving information on all aspects of America's largest companies. Compares information on topics including sales per employee, five-year financial performance histories for sales, profits, and earnings per share. Users can take advantage of search and sort options, compare performance of companies and prepare graphs (using five years of data on two companies), and perform custom calculations. Suggested retail price: $99.95.

- *Dell Crossword Puzzles & Other Games* contains hundreds of puzzles, as well as cryptograms and word searches. Suggested retail price: $79.95. An abbreviated version of this product also comes in a floppy disk version with a suggested retail price of $49.95.

- *Money Magazine Financial Assistant* includes calculation templates for evaluating a number of the most common financial decisions. Categories include investments, loans, budgets, and property transactions. Suggested retail price: $99.95.

- *Money Magazine Business Forms* makes it easy to fill out frequently used financial forms, including expense reports, project plans, and loan calculations. It provides twelve templates. This product is delivered on floppy disk; using the optional Newton Connection package, you can download from a personal computer just the form you want to use. Suggested retail price: $39.95.

- *Columbo's Mystery Capers* offers more than 40 brain teasers complete with graphics and text-based clues. Each humorous mystery opens with a text introduction and an illustration of the scene of the crime. Players use the pen to uncover clues. Suggested retail price: $79.95.

AV Macintosh Computers Provide Integrated Telecommunications, Video, and Speech

Since January 1984 when Apple introduced the first Macintosh computer, Apple has brought a long succession of "firsts" to the computing world. The newest Macintosh systems, unveiled right next to the Newton MessagePad at the Macworld Expo in Boston August 3, follow in that tradition, for the first time bringing together telecommunications, video, and speech technologies on a desktop computer.

The Macintosh Quadra 840AV—the most powerful Macintosh yet—and Macintosh Centris 660AV, which were made available worldwide, offer users new and enhanced ways to compute and communicate through a distinguishing set of features Apple Computer, Inc. is calling "AV technologies."

"No other personal computer on the market today can provide all of these capabilities in one integrated system," said Ian Diery, executive vice president of Apple's Personal Computer Division. "With AV technologies, we are extending the Macintosh platform, and providing countless new communications possibilities for our customers and developers. Once again, Apple is raising the industry standard in personal computing."

Designing products that take advantage of the new AV technologies can give you two distinct advantages: First, it gives your customers access to features never before offered together on a desktop computer, further helping to differentiate your products.

Second, Apple eventually expects to incorporate AV technologies into the next-generation PowerPC processor–based RISC Macintosh computers to be delivered starting in early 1994 (although the first PowerPC Macintosh computer will not support all AV technologies). By incorporating the new AV technologies into your products while you prepare them for the PowerPC processor-based computers, you save yourself the development time of doing so later, and give your products a jump on others that don't employ AV technologies. (For technical details about the new AV computers, see "Introducing the New Macintosh Centris 660AV and Macintosh Quadra 840 AV" on page 17 of this issue.)

Technology introduced in the newest Macintosh computers includes the following features:

The Apple AudioVision 14 Display, the first computer display with integrated audio and video capabilities.

- *The GeoPort telecommunications architecture.* GeoPort integrates sending and receiving of data, fax, and voice telephony. GeoPort makes it possible for users to access voice mail, electronic mail, and fax communications from their computers as well as to use the Macintosh as a speaker phone and answering machine.

- *Video in and video out.* These technologies allow users to view videos from VCRs, camcorders, and laserdiscs as well as to display videos and presentations on televisions without any additional hardware. The sophisticated video capabilities of the new Macintosh AV models, when combined with GeoPort and built-in Ethernet networking capabilities, enable a new class of collaborative applications for document sharing and local area network-based videoconference capabilities.

- *Speech recognition and text-to-speech (for now, North American only).* The PlainTalk Speech Recognition system allows users to maneuver through applications by issuing spoken commands, either using standard menu commands or their own customized commands. PlainTalk's Text-to-Speech system provides a standard way for applications to synthesize a human voice, particularly useful for language instruction and proofreading applications.

The AV Macintosh computers employ both a Motorola 68040 microprocessor and an AT&T 3210 digital signal processor (DSP). While the Motorola 68040 microprocessor does general-purpose computing work, the DSP handles specialized tasks and real-time data—including speech, audio, modem, telephone and fax signals. You can also employ the DSP to provide performance enhancements for your products.

New AV Display

To help users take advantage of the new features, Apple is also shipping the first computer display with integrated audio and video capabilities, the Apple AudioVision 14 Display (see photograph on this page). It features an integrated microphone and speaker placement in a unique design (for which Apple currently has several patents pending). Its highly directional microphone, embedded in the front center bezel of the display, uses the same high-quality class of capsule used in lavaliere microphones. The display's stereo speakers were engineered for close-in listening, and they provide superior sound as well as minimal sound spill-over.

The AudioVision 14 Display is ideal for use with the new AV Macintosh computers, but it can also be used with all Macintosh II, Macintosh LC, Macintosh Centris, and Macintosh Quadra computers. Its 14-inch Sony Trinitron screen provides 640 x 480 pixel resolution; the display also offers a power-saver mode that complies with U.S. Environmental Protection Agency Energy Star standards.

Product Specifications

The Macintosh Quadra 840AV is based on a 40-MHz Motorola

68040 microprocessor, as well as a 66-MHz AT&T 3210 Digital Signal Processor (DSP). Expandable from 8 MB of DRAM to 128 MB, the Quadra 840AV features four standard expansion bays for internal storage, including two 3.5-inch bays for hard drives or removable media, and a 5.25-inch bay for removable devices. Offering up to 2 MB of VRAM, the Quadra 840AV supports 24-bit color on displays up to 16 inches and 16-bit color on 21-inch displays.

The Macintosh Centris 660AVV is the highest performance offering in Apple's midrange line. Powered by a 25-MHz Motorola 68040 microprocessor, as well as a 55-MHz AT&T 3210 DSP, the Macintosh Centris 660AV is expandable from 8 MB of DRAM to 68 MB. For internal storage, the Centris 660AV provides one 5.25-inch bay for removable devices. The Centris 660AV offers up to 1 MB of VRAM for supporting 16-bit color on displays up to 16 inches and 8-bit color on 21-inch displays.

Both systems come standard with Ethernet and feature the fastest implementation of NuBus™ ever, NuBus 90. NuBus 90 transfers data up to four times faster than the existing Macintosh NuBus implementations. The Quadra 840AV comes equipped with three high-speed, 13-inch NuBus slots, and the Centris 660AV supports one 7-inch NuBus slot by means of an optional NuBus adapter. In addition, both models support a Digital Audio Video (DAV) connector that provides a high-speed path to and from the raw digitized video and audio data produced by the computer. The DAV slot can be used for real-time audio and video hardware compression cards such as JPEG, MPEG, DVI, and H.261.

The new systems also support 16-bit stereo audio input and audio at various sample rates, including 44.1 kHz and 48 kHz for outstanding quality audio.

Accessories

The Apple PlainTalk Microphone and software that supports North American English voice recognition and text-to-speech will be shipped along with the new AV Macintosh computers in the United States. Outside the United States, the Apple PlainTalk Software Kit and Apple PlainTalk Microphone can be purchased separately.

The GeoPort Telecom Adapter and associated software, with which users can immediately connect to telephone services and support phone calling and answering, as well as send and receive data and fax, will be sold separately. At introduction, the adapter will support analog services at 9600 bits per second (bps).

Announcing AppleSoft Development Products Group

As you've probably heard by now, Apple recently undertook some internal restructuring to adapt to the ever-changing conditions of the personal computer business. In the new world, as it was in the old, developers are important partners for Apple. So we wanted to let you know how these changes will affect our relationship.

Many of the services and products you've received from the Apple Developer Group (ADG) will now be provided by the newly formed AppleSoft Development Products Group, which is responsible for delivering Apple Computer, Inc.'s mainstream development products and services.

AppleSoft Development Products will run the majority of developer support services that used to be administered by ADG, including the Partners and Associates programs, technical support through the Developer Support Center, *Inside Macintosh* and other technical documentation (including *Apple Directions* and *develop, The Apple Technical Journal*), developer tools for Macintosh programmers, APDA, and Developer University.

Evangelism will also remain an imporant part of Apple's commitment to supporting developers' efforts to deliver world-class software and hardware products.

AppleSoft Development Products is headed by newly appointed Vice President Ike Nassi, formerly director of research and technology at Apple's Cambridge, Massachusetts, R&D Center. Dr. Nassi holds a PhD from State University of New York at Stony Brook, and has extensive experience in programming languages and systems, computer architecture, and distributed systems. He's held positions at DEC and Encore Computer, and is currently a research affiliate of the Massachusetts Institute of Technology.

As usual, *Apple Directions* will continue to keep you posted about changes at Apple and how they affect you.

Apple Ships AOCE Final Beta-Test Version to Developers

With the release of the first AOCE products just around the corner, Apple Computer, Inc., has sent a final "beta-test" version of AOCE software to developers worldwide. This final seeding before the retail release includes beta versions of the first two Apple products based on AOCE (Apple Open Collaboration Environment), PowerTalk client (user) software and PowerShare Collaboration Services software, as well as technical documentation and other materials you'll need for building AOCE capabilities into applications.

AOCE is the extension to Macintosh System 7 that provides a platform for integrating all forms of communication and collaboration into the Macintosh user experience. In doing so, AOCE offers a foundation for team productivity solutions; it also provides capabilities for developing software that automates "workflow," a current buzzword that refers to the movement of paperwork around an office.

Today's computer users demand the ability to communicate easily and to work together regardless of the communications or networking facilities they have. But currently available communications and collaboration capabilities are implemented at the application level rather than at the system level, requiring users to learn various ways of using these capabilities and requiring huge efforts from developers to deliver them. AOCE changes all that, providing a consistent, system-level communications infrastructure for facilitating collaboration among users.

If you want to build collaboration and communications features into your application, AOCE is going to make things a lot easier. Instead of having to labor long and hard on the messaging, directory, and security infrastructures that underlie collaborative tools, you can now spend your valuable development time incorporating desirable new features into your products by using AOCE managers and packages. These new facilities provide standard user and programming interfaces for access to mail, messaging, catalog, authentications, and digital signature services.

We've Got the Power to Be Open

With the introduction of PowerTalk and PowerShare (see the AOCE news article on page 11), Apple Computer, Inc., now has no fewer than six technologies/products using names with the word *Power* (not to mention two that use the word *Open*). To help you sort things out, we're providing the accompanying list, suitable for clipping out and pinning to your bulletin board (or even framing). And, yes, there will be a quiz.

OpenDoc Apple's open, cross-platform architecture for compound documents—that is, documents with text, graphics, QuickTime content, and other kinds of data. OpenDoc replaces today's large, single-purpose applications with collections of "parts," each of which consists of one kind of content—for example, a spreadsheet, styled text, or a graph.

PowerBook Apple's line of portable, laptop Macintosh computers.

PowerCD Apple's three-in-one CD-ROM drive. It plays CD-ROM titles on a Macintosh computer, enables viewing of Kodak Photo CD images on a television or a Macintosh, and plays audio CDs through a stereo system, headphones, or Macintosh.

PowerOpen The successor to Apple's current "open systems" operating system, A/UX. PowerOpen is a new version of UNIX® under joint development by Apple and IBM, essentially combining the Macintosh user interface with IBM's AIX. PowerOpen is an alternative operating system for PowerPC processor–based Macintosh computers, and is also designed to run other companies' PowerPC processor–based computers. Existing and future 680x0 and PowerPC Macintosh applications will run without modification on any computer using the PowerOpen operating system.

PowerPC The new high-performance RISC (reduced instruction set computing) microprocessor family being developed by Apple, Motorola, and IBM. Apple will use PowerPC chips in its next generation of Macintosh computers, eventually replacing the Motorola 680x0 chips currently being used throughout the product line. Current applications for 680x0 Macintosh computers will run without modification in emulation mode on PowerPC processor–based Macintosh computers, although Apple is urging developers to translate or port applications so that they can take advantage of the power and new features of the next-generation computers. PowerPC processor–based Macintosh computers will ship with Macintosh System 7.

PowerTalk AOCE system software for Macintosh and PowerBook computers. PowerTalk gives users access to AOCE collaboration services, including a "universal mailbox" for integrated electronic correspondence, an easy drag-and-drop method of sending documents, a catalog capability for access to all kinds of information, and a "digital signature" capability, which enables electronic approval of documents.

PowerShare AOCE server software providing server-based collaboration services for AppleTalk networks. Users of Macintosh computers connected to an AppleTalk network can access these services using PowerTalk client software. PowerShare services include a shared store-and-forward facility for electronic messaging, server-based authentication and privacy for secure network communications, and centralized administration of shared catalogs that are distributed across multiserver networks for system scaling with optimal performance. ♣

For example, the Standard Mail Package provides programming interfaces to a standard user interface that enables any application to send a document, enclose other documents, and sign letters with minimal investment on the part of the developer.

Apple Evangelist Michael Bayer describes how you can best take advantage of AOCE: "With AOCE, we're introducing an entirely new category of software. Until now, we've heard a lot about groupware collaborative applications, but they're generally limited to a small group of people working over a local area network on a particular project.

"People in the workplace, however, are always part of a multitude of teams contributing to a variety of different projects. Today personal computer users are forced to use many disparate facilities to participate in these teams. They communicate over different types of networks—peer-to-peer, client-server, host-based, networks that depend on telephone connections, and so on.

"AOCE will let developers deliver truly collaborative, user-focused applications that enable people to work together on a wide variety of types of projects independent of communication infrastructure," says Bayer.

In addition to making your existing applications "mail capable" through support of the Standard Mail Package, Bayer identifies a variety of specific new opportunities AOCE makes possible that you'll want to consider pursuing. Here are some of those opportunities:

- *Agent software.* This software includes agents that acquire information (by monitoring information sources) and agents that deflect messages to their proper destinations, thereby saving time.

• *Workflow solutions.* Workflow solutions increase productivity by making the processes of routing and handling paperwork simpler and less expensive. (Such solutions can have an immediate, measurable effect on a company's profits by making office workers more efficient.)

• *Telephony applications and telephony-aware applications.* These applications use telephony Apple events to place calls and take actions based on telephone activity.

• *AOCE Template development.* Through AOCE Templates, Apple is for the first time providing a mechanism for adding new functionality to the Finder.

Apple Directions will give you more details about how you can take advantage of AOCE after its formal release this fall; until then, you can refer to the technical documentation and other information on the AOCE Beta July 1993 CD-ROM. If you haven't received yours yet, contact the Developer Support Center in the United States (phone: (408) 974-4897; AppleLink: DEVSUPPORT) or your local Apple office outside the United States. Consult the AOCE Talk bulletin board on AppleLink (path—Developer Services:Developer Support) for the latest AOCE software updates, sample code, and late-breaking news.

Developer University Moves, Offers Discount on Self-Paced Courses

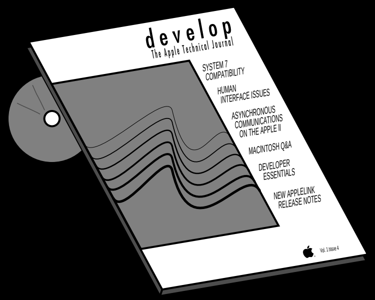
With the completion of the new Apple Computer, Inc., R & D campus in Cupertino, Developer University has moved its training facility—it's now located in the new R & D 1 building. DU's new address is Developer University, Apple Computer, Inc., 1 Infinite Loop, M/S 305 1 TU, Cupertino, CA 95014; phone (408) 974-4897. Drop by and see DU's state-of-the-art classrooms.

In addition, the special offer on Developer University's self-paced training courses continues through the end of September. Until then, for $395 you can purchase through APDA either the Macintosh Programming Fundamentals self-paced course (regularly $595) or the Intermediate Macintosh Applications Programming self-paced course (regularly $495). Or, for a savings of $695, buy them both for the special bundle price of $695. For APDA ordering information, see page 36. (Note that all prices are for the United States.)

Also, watch *Apple Directions* for news about forthcoming courses on Newton programming, to be offered beginning this fall both in self-paced and instructor-led formats.

OpenDoc AppleLink Address Correction

If you want to get started developing for OpenDoc, send an AppleLink message with a description of your idea to AMBER.IDEAS. (On page 5 of the August issue, we reported an incorrect AppleLink address for OpenDoc; we apologize for any incovenience this may have caused you.) For details about OpenDoc, see Gregg Williams's Strategy Mosaic, "Why 1994 *Will* Be Like 1984," in the August *Apple Directions.* ♣


develop
The Apple Technical Journal
SYSTEM 7 COMPATIBILITY
HUMAN INTERFACE ISSUES
ASYNCHRONOUS COMMUNICATIONS ON THE APPLE II
MACINTOSH Q&A
DEVELOPER ESSENTIALS
NEW APPLELINK RELEASE NOTES

Technology

Inside This Section

develop Issue 15: QuickDraw GX and Much More

Issue 15 of *develop, The Apple Technical Journal,* is loaded with articles about QuickDraw GX—but that's not all! If you want to know more about components, floating windows, or 3-D interfaces, don't miss this issue. Here's a quick look at what you'll find:

- "Getting Started With QuickDraw GX" provides a brief introduction to QuickDraw GX, along with a simple GX-aware sample.
- "Developing QuickDraw GX Printing Extensions" tells all about these nifty new add-ons to QuickDraw GX printing.
- "QuickDraw GX for PostScript Programmers" compares the two graphics models and provides useful code snippets for each.
- "Managing Component Registration" tells how to manage the component registration process for those cases in which it's necessary.
- "Dynamic Customization of Components" describes a sample derived media handler that "speaks" the text track in a movie.
- "Floating Windows: Keeping Afloat in the Window Manager" provides a way to implement floating windows without patching traps, and a library you can use in your own application.
- "Working in the Third Dimension" shows off a nice 3-D interface and presents a set of MacApp objects that you can use to create your own such interface.

On a lighter note, this issue introduces a new column, "View From the Ledge," that teaches office survival skills for the socially and politically inept. There's also Dave Johnson's thoughts on symmetry, a puzzle about a server crash from KON and BAL, and the ever-popular Q&A section.

So be sure to check out *develop* on this month's edition of the Developer CD Series or on paper if you subscribe to the printed copy. The code described in the articles is on the CD, as are all its back issues. We think there's a lot to like about *develop* in general, and Issue 15 in particular; let us know what you think by sending an AppleLink message to DEVELOP. ♣

Caroline Rose,
Editor, develop

CD Highlights

Reference Library Edition September 1993: *Spinal Trap*

Hello and welcome to the September Reference Library Edition of the Developer CD Series, *Spinal Trap,* featuring over 120 MB of new and revised technical documentation, tools, utilities, and system extensions. In addition to a new batch of Macintosh and ESD Tech Notes and Developer Notes, this month's CD features the following new material.

ASLM 1.1GM

Contains the Apple Shared Library Manager 1.1GM (ASLM) development environment for Dynamic Linking Libraries (DLLs) and Shared Libraries development. This release features dynamic linking, loading and unloading of code, and shared libraries functionality with support of C++ and procedural C (call by function name).

ColorSync 1.0.3

ColorSync is a system extension that enables consistent color across the system. ColorSync allows you to display and print colors as they appear in the original art you scan.

ColorSync 1.0.3 is an upgrade release for the support of both QuickDraw and QuickDraw GX, and features these enhancements:

- improved color-matching performance in Apple default color matching module (CMM)
- support for CMYK input in the CWMatchColors routine
- new ColorSync profiles for PowerBook 165c, 180c, and Macintosh LC 520
- improved support for 68040-based Macintosh computers

Developer Notes

Included here, along with our regular archive, are developer notes for several new products, including the AudioVision 14 Display, Macintosh PowerBook 165, Macin-

Spinal Trap

tosh Centris 660AV, and Macintosh Quadra 840AV.

Hardware System Update 2.0

This update offers software improvements to meet the ongoing needs of Macintosh users as well as enhancements to address specific problems. It requires Macintosh system software version 7.1 and is not intended for Macintosh Performa users. The update

- addresses possible floppy disk and hard disk problems for some users of the Macintosh IIsi, Macintosh IIci, Macintosh IIvx, Macintosh IIvi, Macintosh Centris, Macintosh Quadra, and PowerBook computers
- manages fonts in the computer's memory more efficiently
- improves the accuracy of the system's clock in Macintosh LC, LC II, IIsi, IIvx, IIvi, Macintosh Classic II, Macintosh Quadra 900, and Macintosh Quadra 950 computers
- corrects problems ejecting floppy disks when you shut down Macintosh IIsi, IIci, IIvx, IIvi, Macintosh Quadra 700, and Macintosh Quadra 950 computers
- corrects color problems users may have with the Apple Basic Color monitor on Macintosh Centris and Macintosh Quadra computers
- removes horizontal white lines that may appear while scrolling on Macintosh Quadra and Centris systems with a Macintosh 16" Color Display
- improves the reliability of high-speed modems and similar devices connected to Macintosh LC, LC II, IIsi, IIvx, IIvi, Macintosh Classic II, Macintosh Quadra 900, and Macintosh Quadra 950 computers
- includes HD SC Setup version 7.2, which gives users more flexibility to format large hard disks that store between 1 and 2 gigabytes of information
- includes Apple Disk First Aid version 7.1.1, a utility that repairs damaged disks
- updates the PowerBook control panel and Express Modem software to support standard communications software
- includes Sound Manager version 3.0, which enhances sound performance on most Macintosh computers
- adds software for the Apple Adjustable Keyboard to allow users to connect the keyboard to any Macintosh computer without separately installing special keyboard software
- updates the Memory control panel (version 7.1.1), fixing memory problems that may occur while large RAM disks are used with System 7.1
- updates the System Enabler (version 1.0.5) for the Macintosh Color Classic, which enables users to add the Apple IIe Card and software to the Classic and improves mouse performance on the Color Classic

LaserWriter Pro Energy Star Software

The LaserWriter Pro Software Upgrade allows the LaserWriter Pro 600 and 630 printers to save energy, meeting the requirements of the Environmental Protection Agency's Energy Star program. Once users install this upgrade on a LaserWriter Pro 600 or 630 printer, the printer meets Energy Star requirements.

Macintosh Technical Notes

Technical Notes are a collection of short (and not-so-short) articles dealing with specific development topics. This month features new and updated Tech Notes and Q&As for September 1993. For fans of the old numbering system, the folder "!Indices: Old Numbering:" contains a list of aliases to the Notes sorted by old number and title.

Network Software Install Version 1.3.3

This Network Software Installer (NSI) is for AppleTalk version 58.0.2. It includes updates to the EtherTalk and TokenTalk drivers, A/ROSE 1.2, which allows the Ethernet NB card to work on the Macintosh IIvx system. NSI 1.3.3 correctly installs AppleTalk version 58 to a PowerBook Duo system, and it can install Ethernet software on the Macintosh Quadra 800 and Centris 610 and 650 systems.

ODBC for Macintosh

Open Database Connectivity API (ODBC) is a client-server programming interface that provides a standardized interface to relational data. ODBC was originally created by Microsoft for the Windows environment, and, through a joint project between Apple and Microsoft, is now being provided by Apple for the Macintosh environment. By providing ODBC in both the Macintosh and Windows environment, Apple and Microsoft are providing a consistent cross-platform API for relational data access.

Sound Manager 3.0

Available since the introduction of the Macintosh II in 1987, the Sound Manager is Apple's digital audio software solution that allows any application to play and record sounds using the built-in sound hardware found on Macintosh computers.

Sound Manager 3.0 is the latest version of this software. It adds support for 16-bit CD-quality audio, redirection of sound to third-party hardware cards, and plug-in audio compression/decompression software (codecs). It is completely backward compatible with previous versions of the Sound Manager.

Telephone Manager 1.1.1

The Telephone Manager is the cornerstone of the Macintosh Integrated Telephony Architecture. The Telephone Manager abstracts telephone service from telephony applications.

Telephone Manager 1.1.1 is an update containing various bug fixes. The previous Telephone Manager release, version 1.0, was contained in the Telephone Manager Developer's Kit, available through APDA. Developers should refer to this package for Telephone Manager documentation.

On tap for next month: a new beta version of QuickDraw GX, the *Human Interface Guidelines* in Apple DocViewer, new localized version version of System 7.1, the long-promised network game demo, and more. See you there!

Alex Dosher
Acting Developer CD Project Leader

develop Wins International Award

We're happy to announce that *develop, The Apple Technical Journal* won an Award of Excellence in the 1993 International Technical Publications Competition sponsored by the Society for Technical Communication. ♣

Human Interface

The Eyes Have It

By Peter Bickford

This month, I thought I'd reach into the mailbag and see what was there. Perhaps the most intriguing letter I found was the following one.

Dear Doc,
I am in the process of converting some applications from OS/2 and Windoze to the Mac (OK . . . I've seen the light!), and I want to be certain that I've got the Human Interface right the first time . . . . The apps to be converted all have a window structure roughly as follows:

```
 ______________________________________________________________
| Status area (fixed vertically, but autoscrolls horizontally) |
 ______________________________________________________________
| Variable area (e.g. History log)                  | |
|      with vertical scroll bar                    | |
|                                     | |
|                                     | |
|                                     | |
|                                     | |
|                                     | |
|                                     | |
 ______________________________________________________________
| Input area (fixed vertically)            | Button | |
 ______________________________________________________________
| Message area (fixed vertically)                 |
 ______________________________________________________________
```

My first reaction is to make the window fixed in size, and use a dialog box for any messages. However, I'm not certain what the guidelines say about the input area. . . . I assume it should be a box, but do the guidelines say anything about whether it should be at the top or bottom, or floating or what? It is inappropriate to input directly to the variable area, because that will be accumulating input from local and remote systems, and interleaving them to give a timed historical record of activities.

This may seem like a very simple question, and I apologize for troubling you with it . . . however, as a lone Mac developer surrounded by OS/2, AIX, and Windoze bigots (actually there are not many of the latter here!), I would appreciate any help and guidance.

Regards,
Douglas Shiell
IBM United Kingdom Laboratories Ltd.

Hi Douglas,

There are two basic questions at stake here: where most stuff goes, and where message stuff goes. Let me take these questions one at a time. Naturally, I'll feel obliged to pontificate a bit en route, but if you stick with me, your questions will be answered.

Follow the Reading Path

I'll first take on the question of where stuff goes in general. In most Western languages, reading goes from left to right, from the top of the page to the bottom.

You should note that other languages, such as Hebrew and Chinese, are read in different directions, such as right to left/top to bottom, or top to bottom/right to left. If you're doing screen design for such a system, you'll need to adjust what is said here accordingly.

Whenever possible, the user's "work" should follow this reading path. For example, a loan officer's job involves looking over a client's credit history, then making a decision and entering information based on what he or she saw. It makes sense, then, to try to locate the client's credit history toward the top of the screen, and to put the area where the loan officer enters comments or approval amounts toward the bottom. If the loan officer needs to press a button to confirm or cancel input, those buttons should go to the right or below where the input was entered.

In your case, whether the input area should go on the top or the bottom depends mostly on whether users need to read the information in the scrolling history window before entering data. If they do, you'd do well to locate that area toward the bottom of the screen. If not, put the input area at the top. If the history data has no bearing on what is being entered, you might also consider putting it in a separate window altogether.

Tunnel Vision

Everything I've said up until now relates to things people actually read. Of course, this means that what I've said so far has little or no bearing on a computer program's status messages, or even its dire warnings of impending disaster.

The culprit is our lazy-butt good-for-nothing perceptual system. Here we go to all the trouble to provide it with an interesting world, full of countless things to look at, smell, touch, taste, and hear, and it ignores almost all of it. It's constantly cutting corners, trying to fake us out by only paying attention to the "important" stuff, and utterly disregarding just about everything else.

Take driving, for instance. As we speed down the highway, we could be paying attention to all sorts of neat things, from the feel of the upholstery to the sonic textures of the song playing on the car stereo. Instead, our ergophobic senses can't be bothered to do anything except keep us alive by concentrating on a couple of measly cars zooming around straight in front of us. For as much as our eyes watch anything else, we might as well be driving down a tunnel (hence the term "tunnel vision").

Driving isn't the only time we get tunnel vision. When we read a book, we almost never read the "unimportant" parts of the page, such as headers and page numbers. And when we look at a computer screen, we concentrate almost all our visual efforts on the current window.

But even that's not enough for our layabout perceptual system. It's constantly on the lookout for "visual dead spots" of the window, which it can safely ignore most of the time. Dead spots can occur anywhere the user doesn't usually find items of interest, but a tell-tale sign of a dead spot is a line separating the spot from the rest of the window. Thus window titles and scroll bars are usually visual dead spots. The "status" and "message" lines in your diagram are other likely examples.

Knowing how to use visual dead spots is a key skill in window design. By consistently putting controls in the same part of the window, we allow users to learn where they are but safely ignore the controls until they are needed. That way, we give users the comfort of feeling that they always know where the controls are (not that they actually look half the time!) and, at the same time, we don't tax their visual sense with what usually is superfluous information.

Warning! Warning!

The danger comes when you put something you want users to notice in a dead spot. Because users are accustomed to ignoring the information there, you may have to resort to bright colors, flashing, alarm klaxons, or other equally annoying measures to get people to wake up and pay attention. Even then, the results can be uneven.

A better approach than the "status line/message line" is to make appropriate use of cursors and alert boxes. Cursors are great for lightweight "I'm busy/I'm done now" sorts of messages. Alerts are better when the user's life or mental stability is likely to be affected if the message is not received. Both of these also have the virtue of being highly noticeable.

I would also caution against using some area of the screen (such as the lower-left corner of the window) as an ongoing "help line." In most cases, the user will have no interest in the information being displayed, and the information will, at best, be ignored. A more serious consideration is that the flickering of different messages on the help line may distract users, drawing their limited visual attention away from their work.

A better solution, for all its faults, is Balloon Help. Then, the "help" is displayed at the same place users are working, so their eyes don't need to dodge between their work and the help line. More important, it's only on when users want it on. Given the sad state of our perceptual system, the less information you burden the user with, the better.

Till next time,
—Doc
AppleLink: THE.DOKTOR

Having survived his sabbatical leave, Pete Bickford is still a member of the Apple Enterprise Systems—now called Apple Business Systems—human interface team.

Introducing the Macintosh Centris 660AV and Macintosh Quadra 840AV

By Amr Eissa

The Macintosh Centris 660AV and Macintosh Quadra 840AV computers, released August 3 at the Macworld exposition in Boston, bring a new world of audio and video capabilities to Macintosh developers and users. The newest Macintosh computers combine significant advances in telecommunications, video, speech, and sound into a single package Apple calls "AV technologies" that maintains ease of use and cost effectiveness.

These advances are possible because of new application-specific integrated circuits such as the Cyclone Integrated Video Interfaces Controller and the Direct Memory Access (DMA) controller. The Cyclone Integrated Video Interfaces Controller controls video memory and the display of video signals and Macintosh graphics. The Direct Memory Access controller handles the transfer of data to and from memory. Computer performance increases because the microprocessor can perform other tasks while the custom integrated circuits control the display of graphics or the transfer of data.

Other new technologies featured in the AV Macintosh computers include a built-in digital signal processor (DSP), which is a special processor designed for high-speed numerical processing. Interpreting modem signals, performing speech recognition, and smoothing and manipulating images are some of the functions performed by the DSP. The AV Macintosh computers also include several new application-programming interfaces (APIs), such as the Apple RealTime Manager, which controls the functions of the DSP, and the Speech Manager, which creates synthesized speech.

The Centris 660AV includes the major features of the Macintosh Quadra 840AV with some important differences. The Centris 660AV has a 25-MHz 68040 microprocessor and a 55-MHz AT&T DSP3210 digital signal processor, while the Macintosh Quadra 840AV uses a 40-MHz 68040 microprocessor and a 66-MHz AT&T DSP3210 digital signal processor. The Centris 660AV has a maximum of 68 MB of 70-ns DRAM and 1 MB of video memory; the Macintosh Quadra 840AV has a maximum of 128 MB of 60-ns DRAM and 2 MB of video memory. The Centris 660AV has a processor-direct slot that accepts a 7" NuBus board via an adapter, while the Macintosh Quadra 840AV has three NuBus slots but lacks a processor-direct slot.

Developer Advantages

It's important to factor the AV Macintosh computers into your near- and long-term development plans for the following reasons:

- Apple is introducing the AV technologies in answer to market demand for video display and editing, speech recognition, and telecommunications features. Applications that take advantage of these technologies will differentiate themselves in the marketplace and be more attractive to today's and tomorrow's customers, especially those that require the AV feature set.

• Apple intends that, eventually, the features of the Macintosh Quadra 840AV and Centris 660AV will be adapted to the PowerPC processor-based Macintosh computers, although it's important to note that not all AV technologies will be available with Apple's first PowerPC computers. By developing AV-savvy applications, you get a head start on developing applications for the PowerPC processor-based Macintosh systems. Applications that take advantage of the new technologies will have a significant marketing advantage when PowerPC processor-based computers with AV technologies become available.

• After the introduction of the PowerPC processor-based systems, there will still be a broad base of customers who need applications that run on 680x0 Macintosh computers. The Macintosh Quadra 840AV and Centris 660AV provide excellent platforms for developing 680x0 Macintosh products.

• The Macintosh Quadra 840AV and Centris 660AV—which employ a special hardware and software architecture called the GeoPort that works like a built-in modem—target users who require the integration of telephone, video, and computing capabilities. An example of such users includes the increasing number of people, especially in office settings, who want to do videoconferencing. You can now build applications that employ new telephony technologies to reach these users.

• There's increasing demand to edit, save, and display video images on computers; the new AV Macintosh computers tap into this demand, since they can be connected to televisions (for display) and VCRs and camcorders (for capture of video data) without any additional hardware. However, video data takes up huge amounts of storage space, so the introduction of these new technologies also opens a special opportunity to developers of video compression and decompression cards.

The technology introduced in the Macintosh Quadra 840AV and Centris 660AV provides both opportunities and challenges for you. Learning new APIs, developing new features, maintaining compatibility with older Macintosh systems, and preparing for the upcoming PowerPC processor-based systems is a mixture of excitement and hard work. It takes time and energy to learn new skills, but the end result is a product that does something better than ever before.

What's New With Macintosh Quadra 840AV and Centris 660AV

The following paragraphs describe the features that distinguish the AV Macintosh computers from earlier Macintosh systems.

Digital Signal Processor. The digital signal processor (DSP) works as an extra computer in the Macintosh Quadra 840AV and Centris 660AV. The DSP contains special instructions and circuits for performing mathematical operations rapidly on large groups of numbers. The ability to rapidly process numbers becomes critical to computer performance when it comes to processing objects like pictures, sound, or modem signals.

For example, suppose you want to smooth a computer image or send a file over a modem. The computer must first convert photographs to numbers representing shades of gray or colors, or modem signals to numbers corresponding to changes in voltage signals transmitted over a phone line. Next, the CPU must process the data by performing many operations on these large groups of numbers. Normally the CPU has to do all of the work of processing the data. Computer performance suffers because the CPU is unable to do other things during this time. The DSP takes over these processing tasks, freeing the CPU.

The Centris 660AV and the Macintosh Quadra 840AV use the AT&T DSP3210, which contains its own instruction set and operating system. The DSP operating system controls how the DSP processes data and optimizes processing based on the type of data.

The Macintosh Quadra 840AV and Centris 660AV are the first Macintosh systems to include a built-in DSP. Many key features rely directly on the DSP. Here are some of the new and enhanced features provided by the DSP:

- the speech recognition portion of PlainTalk (called Casper at the 1992 and 1993 Worldwide Developers Conferences)
- software simulating a 14,400 bps modem
- specialized programs that speed up image processing for graphics applications
- playback and manipulation of 16-bit sound

You can use the DSP to enchance application performance in a variety of ways, without knowing anything about the DSP (unless you choose otherwise). You access the DSP automatically when using the Sound Manager and Communications Toolbox to perform sound, telecommunications, and speech recognition functions. When you need more features than those provided by the Sound Manager or Communications Toolbox, you can program in C or C++ using the routines included with the RealTime Manager. The RealTime Manager provides direct access to DSP features through the Macintosh Toolbox.

If you really need the speed that comes from maximal optimization of DSP routines, you can program the DSP using DSP assembly language. To help you program the DSP, Spectral Innovations, Torques Systems, and others will soon be delivering DSP tools and compilers.

Video. The Macintosh Quadra 840AV and Centris 660AV support input of NTSC, PAL, or SECAM signals and output of NTSC and PAL signals. Video sources may be either composite video (standard yellow RCA jack) or S-video (4-pin mini-DIN connector) devices.

The so-called integrated video interfaces controller manages video random access memory (VRAM), memory dedicated to storing graphics or video, in a very flexible way. It divides the available VRAM into two functions: one to store Macintosh graphics, which includes screen displays, QuickTime movies, and applications windows; and one to store video images from an external source. The video controller reorganizes VRAM to store only graphics when the computer displays graphics using the highest number of colors. The ability to display video sources is disabled until the user reduces the number of colors in the Monitors control panel.

With video display enabled, the video controller transfers video data to a portion of VRAM used as a frame buffer. Frame buffers improve video performance because no other graphics are stored in that section of VRAM during video display. Storage of new video frames in the frame buffer occurs while transferring the current video data to the monitor. Rapid data transfer results in full-screen, flicker-free display of video images.

QuickTime capture applications perform frame capture, and processed video images can be printed to tape or displayed on any composite or S-video source. A video decompression card, installed in the new Digital Audio/Video (DAV) connector, takes compressed data from a hard drive or CD, decompresses the data, and sends it to the frame buffer. Video images from the frame buffer can be compressed and sent to a hard drive or a NuBus card when using a compression card in the DAV connector.

A VCR, camcorder, or laserdisc player can connect directly to the Macintosh Quadra 840AV and Centris 660AV as an I/O device. A significant market exists for video applications, since there are approximately 90 million VCRs and 70 million camcorders in the United States. Displaying a video image from a VCR or saving a QuickTime movie to videotape requires no additional hardware. Customers no longer need to buy interface cards for their VCRs or digital capture cards for making QuickTime movies. As we've already said, the AV technologies should increase demand for video compression or decompression cards, since users will have a greater need to store video data on hard drives.

Speech. The Macintosh Quadra 840AV and Centris 660AV implements PlainTalk (Casper), which includes speech recognition and text-to-speech. Speech recognition involves recognizing and responding to verbal commands from the user; the feature currently requires a Macintosh with a DSP. Text-to-speech, performed by the CPU, allows the computer to talk to the user in a natural-sounding voice; it currently runs on all Macintosh systems.

Apple events and AppleScript make up the interface between applications and PlainTalk for speech recognition. Sets of speech rules, created with AppleScript, control menu items and options in dialog boxes, along with most other commands. Synthesized voice created by the PlainTalk text-to-speech system uses 1000 times less storage space than digitized voice. The Speech Manager controls speech synthesis by converting strings to words, and the Sound Manager controls output through the DSP. Once again, the Sound Manager automatically accesses the DSP functions. The Sound Manager produces sounds on Macintosh systems without a DSP by maintaining backward compatibility with previous versions of the Sound Manager.

You gain benefits from PlainTalk because speech provides a natural extension to the Macintosh operating system. Applications using speech recognition for command entry provide customers with a highly attractive alternative to using the keyboard and mouse. Other uses of speech recognition include verbal database inquiries and interactive educational programs. Support for dictation, a more complicated function than command entry, will occur in later versions of PlainTalk. Current applications of text-to-speech include proofreading and language instruction. Future uses include accessing electronic mail or faxes over the phone with the Speech Manager reading messages to the user.

Telecommunications. A new hardware and software architecture called *GeoPort* enhances telecommunications functions. GeoPort, a family of communications devices, consists of a GeoPort telecom adapter, circuits on the logic board, modem software, and GeoPort software.

GeoPort allows connection to analog telephone, PBX, and ISDN lines anywhere in the world. The GeoPort telecom adapter provides a simple interface between the computer and the telephone system where customers purchase the adapter specific to their communications system. Apple is developing an adapter for analog telephone systems that will be sold separately, and third parties will develop adapters for ISDN (all digital) or PBX (analog and/or digital) support. The Macintosh Quadra 840AV and Centris 660AV are the first computers to emulate a built-in 14,400-bps data and fax send/receive modem as software. No hardware modem exists and the DSP performs all modem functions.

The introduction of the GeoPort architecture opens opportunities to create specific GeoPort adapters for different communications systems. The Telephone tool, available later this year from Apple, together with PlainTalk will open further opportunities in the telecommunications area: the creation of applications that allow customers to access a remote computer and retrieve electronic mail over the telephone.

Full Direct Memory Access (DMA). The Macintosh Quadra 840AV and Centris 660AV are the first Macintosh systems to implement full direct memory access. With direct memory access, data transfers occur at high rates between random access memory and disk drives, Ethernet, or serial ports without the supervision of the CPU. Normally, the CPU controls data transfers between memory and storage devices, and performance suffers because the CPU cannot perform other tasks during this time. This becomes a greater problem when transferring large files, such as those including sound or images. DMA improves computer performance, since data from a hard drive goes directly to memory under the supervision of the DMA controller leaving the CPU free to perform other tasks.

Sound. The Macintosh Quadra 840AV and Centris 660AV provide increased support for sound. Sixteen-bit, CD-quality stereo sound can be recorded, stored, and manipulated. The DMA controller transfers sound data to the DSP, which then performs audio compression/decompression, effectsprocessing, or playback. Sampling rates up to and including 44.1 kHz and 48 kHz are supported.

Uses for the new sound capabilities include video authoring and editing, education, and entertainment. You use the new Sound Manager 3.0, which represents a major improvement over previous versions of the Sound Manager, when implementing the new features.

Asynchronous SCSI. SCSI Manager 4.3 improves data throughput by supporting asynchronous operations where data is transferred with minimal control by the microprocessor. More than one SCSI device may operate at the same time: Devices disconnect from the bus while they process commands, allowing other devices to access the bus for data transfer. SCSI Manager 4.3 makes use of direct memory access on the Macintosh Quadra 840AV and Centris 660AV providing increased performance during data transfers. Maximum data transfer rates of 5 MB/second can be achieved with the new SCSI Manager. For further details about SCSI Manager 4.3, see "Apple Announces Asynchronous SCSI Manager" in the June 1993 *Apple Directions*.

Improved NuBus Controller. The new NuBus controller enables data transfer rates four times faster than that of NuBus controllers on previous Macintosh Quadra computers. With the new controller, data transfer rates between the CPU and NuBus card are 40 MB/second (10 MHz x 4 bytes). Between NuBus cards, data transfer rates are 80 MB/second (20 MHz x 4 bytes).

How to Take Advantage of the New Features

The AV Macintosh computers' new features are based on a number of APIs and technologies. A knowledge of each of these areas is important for you to develop applications that take advantage of the new features of the Centris 660AV, the Macintosh Quadra 840AV, and the PowerPC processor-based Macintosh systems. To use the full feature set of the Centris 660AV and the Macintosh Quadra 840AV computers, your applications must support the following technologies:

- *System 7.1:* Your applications should be compatible with System 7.1 to support the new features and avoid problems with the new CPUs.
- *Apple events:* Speech recognition and scripting are just two of the new features that are built on Apple events. These features require full support for Apple events. Apple events and the *Apple Event Registry* are included with the May 1993 Developer CD and E.T.O: Essentials•Tools•Objects CD #11, May 1993, pathname Essentials:Apple Events Dev. Toolkit. E.T.O, available from APDA, contains a complete set of Apple development tools including the Macintosh Programmer's Workshop, MacApp, MPW C and C++ compilers, plus system software, sample code, and technical documents. (For APDA ordering information, see page 36.)
- *AppleScript:* AppleScript, a user-oriented programming language, supports voice recognition, speech control of applications, and automation of complex tasks that involve multiple applications. Apple events must be supported in your application if they are to be scriptable. AppleScript is available on E.T.O #11, May 1993, through the path Essentials: AppleScript Setup.
- *Asynchronous SCSI:* The new SCSI Manager 4.3 is backward compatible with earlier versions of the SCSI Manager. Applications have to be modified, however, to take advantage of the new features, such as disconnect/reconnect and asynchronous SCSI. You must use the SCSI Manager 4.3 for applications and drivers, since bypassing the SCSI Manager results in errors. The August 1993 Developer CD contains SCSI Manager 4.3.
- *GeoPort:* The GeoPort is backward compatible on all Macintosh systems. The Communications Toolbox is used to support fax and modem functions, and the Telephone Tool will support telephone or answering telephony applications. The Communications Tools are available from APDA as the Macintosh Communications Tools Basic Connectivity Set, version 1.0. E.T.O #11, May 1993, contains the Communications Tools in the path Tools-Objects:N&C Tools:Communications Toolbox. The Telephone Tool will be available later this year.
- *QuickTime:* QuickTime 1.6.1 ships with the Macintosh Quadra 840AV and Centris 660AV. Applications must use QuickTime to take advantage of the video input capabilities of the new computers. E.T.O #11, May 1993, contains the QuickTime interfaces through the path Essentials:U.S. System Software:QuickTime 1.6. The QuickTime Developer's Kit from APDA provides a complete QuickTime package for developers.
- *Apple RealTime Manager:* The RealTime Manager provides the interface to the DSP. Applications needing more functions than those provided by the Sound Manager or Communications Toolbox must use the RealTime Manager. Examples of the kinds of applications that might use the RealTime Manager include numerically intensive scientific applications, 3-D rendering, and image manipulation. The Speech Recognition API is not currently available as a separate item. The DSP Tools and compiler, needed to program the DSP in DSP assembly language, will be available from third parties and APDA later this year. The Macintosh Quadra 840AV and Centris 660AV Software Developer Note, available from APDA, describes the features and interfaces of the RealTime Manager.
- *Sound Manager 3.0:* The new Sound Manager, version 3.0, has been rewritten to support 16-bit sound, Macintosh Audio Compression and Expansion (MACE), and Musical Instrument Digital Interface (MIDI). You can control features of the different sound devices by writing to the Component Manager, which implements a control panel specific to the device. The Component Manager Developer Note on the QuickTime CD (versions 1.5 and 1.6) describes how to add functionality to the Sound Manager, and the QuickTime Developer's Kit for QuickTime 1.6.1 contains descriptions of the new interfaces and features provided by the Sound Manager 3.0. A beta version of Sound Manager 3.0 is available on the Worldwide Developer's Conference '93 New Technologies CD.
- *Speech:* Speech recognition allows the user to control computer operations by voice. Speech recognition relies on Apple events and AppleScript to create speech macros to control applications. Text-to-speech relies on the Speech Manager. The August 1993 Developer CD contains a beta version of text-to-speech along with a description of the Speech Manager.
- *Video Input/Output:* The Macintosh Quadra 840AV and Centris 660AV use QuickTime 1.6.1 for video display and video capture. The QuickTime Developer's Kit, available from APDA, contains the tools and information needed to include video support in your applications.

Compatibility

The Macintosh Centris 660AV and the Macintosh Quadra 840AV, based on the 68040 microprocessor and System 7, are highly compatible with earlier Macintosh systems. New features build on a solid platform of proven hardware and software. If you follow Apple's programming guidelines, the applications you design will have little or no compatibility problems when they're run on older Macintosh systems or on the new PowerPC processor-based Macintosh computers. The programming practices detailed in the Macintosh Tech Note "10+ Commandments" (Overview 10) provide the foundation for building a successful application that is compatible with all Macintosh systems.

Here are some specific guidelines you'll want to be aware of when you program for the Centris 660AV and the Macintosh Quadra 840AV computers:

- Applications *must* be 32-bit clean. There is no longer an option for 24-bit memory addressing.
- Don't disable interrupts for more than one millisecond. Applications disabling interrupts for too long a period of time may cause the RealTime Architecture on the DSP to fail.
- Don't access hardware registers directly: Do not program specific hardware registers on SCSI, serial, sound, and graphics subsystems. Applications that access specific registers may have problems since the registers are implemented

differently on the PowerPC processor-based computers.

• Don't write self-modifying code. This will minimize problems when Apple changes memory-protection models in the future.

To repeat some of what's been said in previous issues of *Apple Directions,* here are some specific programming practices to keep in mind when developing for PowerPC processor-based Macintosh systems:

• Align data structures, since the PowerPC processor-based Macintosh systems will be affected more than 680x0 systems by misaligned data structures. Even if the PowerPC processor-based Macintosh supports misaligned data structures, program speed may be compromised.

• Don't depend on 680x0 run-time models, since the run-time model will be different on the PowerPC processor-based Macintosh systems. Methods of accessing registers and calling conventions and stack structure will be different on future systems.

• Don't depend upon interrupt level, supervisor mode, or exception handlers; these features are specific to a given processor.

• Isolate and minimize the use of low-memory values. Many applications depend on low-memory globals, and there is no alternative for this. Isolating and minimizing your use of low memory will decrease your problems should a documented low-memory global change. Do not use undocumented low-memory globals or hardware-dependent low memory.

• Isolate dependencies on 80-bit and 96-bit extended data types, since floating-point support will be different on different platforms. Different floating-point units (FPUs)will have different preferred formats that affect performance and precision.

• Don't depend on resources being in the System file; more of System 7 is being incorporated into ROM. You may need to search both the system file and ROM for a particular resource.

• Don't directly patch the ROM. Use the SetTrapAddress and GetTrapAddress routines to guarantee compatibility.

And, finally, regardless of the kind of Macintosh system you are programming for, here are a few sound, general programming practices you'll want to remember:

• Write your applications in ANSI C or C++.

• Don't depend on undocumented internal Toolbox data structures. If it isn't documented, it's likely to change.

• Use the Gestalt Manager to check availability of system features.

• Don't make assumptions about the ROM size. The Macintosh Quadra 840AV and Centris 660AV use 2 MB ROMs, and the ROM size will increase in the future.

Additional software compatibility issues are discussed in "Message From the Worldwide Developers Conference" in *Apple Directions,* July 1993, and "Macintosh on PowerPC: Top Developer Q&As" in *Apple Directions*, June 1993. ♣

Amr Eissa is president of International Consulting Group located in Saratoga, California.

Apple Evangelizes Client-Server Development Tools

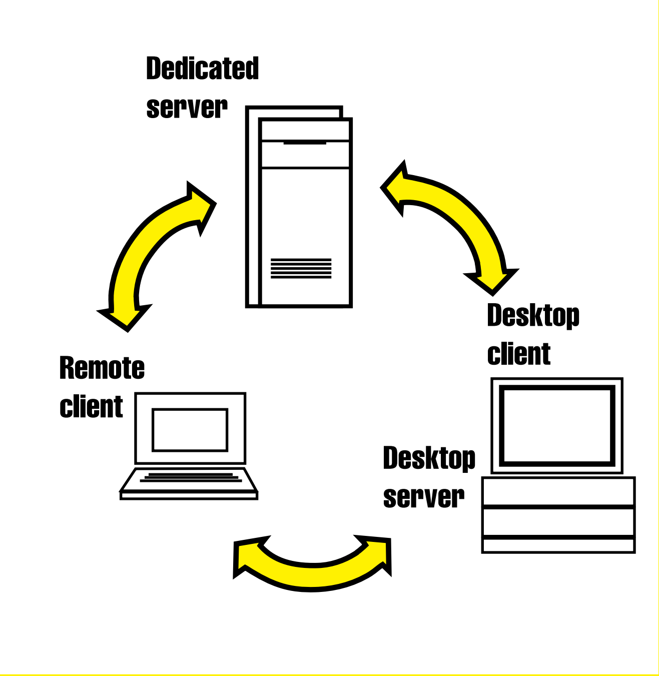
By David Gleason

One of the foundations of Apple Computer, Inc.'s strategic direction is its "client-client-server" architecture (see the front-page strategy articles in the May issue of *Apple Direct* and the June issue of *Apple Directions*). In future products, Apple will continue to provide users with the tried-and-true, familiar, intuitive features for interacting with their Macintosh computers. But at the same time, Apple is extending the user's reach far beyond desktop computers, PowerBook computers, and personal digital assistants (PDAs) to access the world of information sitting on remote computers, regardless of size, shape, and vendor.

Apple's vision is one in which the computing experience consists not simply of a user relating to a computer, but of a user relating, through that computer, to the electronic world. The key is that, regardless of the devices that are accessed, Apple is keeping the user experience the same.

In Apple's vision, the familiar desktop computer is morphed into a redefinable entity; it can function as a client of a larger server, and, at the same time, as the server of another client, such as a remote PowerBook or PDA. This is a computing model that allows for tremendous flexibility and power, giving the user almost unlimited options for creating personal computing solutions.

Apple realizes that flexibility and power is, of course, a challenge for software developers. It's your job as developers to keep the familiar as familiar as



The client-client-server model of computing.

possible—in other words, to design products that adhere to the development and interface guidelines laid out in *Inside Macintosh* and elsewhere—while extending your users' reach to previously unimagined territory. You have to provide products that access data on computers other than the Macintosh, across a myriad of networking architectures and protocols, and with an almost endless array of applications.

Apple is helping developers fill that rather tall order. One significant way we're doing that is through our evangelism of cross-platform, client-server application development tools—a program we feel you will want to know more about. In the past six months, sixteen Independent Software Vendors (ISVs) have announced their participation in Apple's client-server tools partnership program, and there will be more to come in the future.

Why Client-Server Is the New Model

The new direction that Apple has charted has a lot to do with the new realities of business in a world where many companies are downsizing while their data processing needs are growing. Virtually any established enterprise today has a large investment in a combination of mainframes, minicomputers, and personal computers, as well as extensive collections of commercial and in-house application software. So, although computing needs are changing, few can afford to toss out their existing systems and start over.

The sensible solution in this environment is one that allows companies to continue to use their older systems, known as *legacy* systems, and make the transition to newer, more powerful configurations and applications over time. The client-server model makes sense in this regard

Client-Server Development in Action

The following are examples of applications developed with Apple-evangelized client-server tools and AppleScript. Note that AppleScript also can play an important role in client-server development; it's the glue that binds together two of the solutions in these examples.

Gannett Free Press Gannett Co. is the largest newspaper chain in the United States, with more than 80 newspapers, including *USA Today* and dailies across the country. Gannett has developed a high-end publishing solution, called NEWSworks, that it intends to make available for commercial sale. It enables editors to monitor scheduling, track stories and photos, even size articles, requiring very little computer expertise.

By using a variety of commercial applications, such as TouchBase from After Hours Software (recently acquired by Aldus) and QuickKeys from CE Software, Gannett created a customizable database to manage the contacts that writers and editors use for generating and verifying stories and information.

The main tool that Gannett uses is Chena Software's Fair Witness, an information spreadsheet that helps users to outline, schedule, evaluate, and work with ideas and projects. Outlines can be augmented with text, dates, pictures, photos, even QuickTime movies and sound. Fair Witness functions as a project management tool for ideas and information, allowing newspaper editors to consolidate all the components of a daily newspaper automatically.

NEWSworks accesses Fair Witness by means of a script that picks up text, opens TouchBase, performs a series of keyword searches, and then loads sources, addresses, and phone numbers into a separate storage area for future use.

NEWSworks also automates the flow of information from the newsroom, performing many of an editor's tasks, such as individually assigning stories to reporters, downloading text and other story components to the right person's computer, and managing the scheduling according to the paper's deadlines.

Solutions, Inc. Chuck Rogers of Racine, Wisconsin, is a consultant with Solutions, Inc., who creates custom solutions from off-the-shelf packages. He also uses the integration power of AppleScript to automate his own accounting procedures, reducing duplication of effort and improving data integrity.

Rogers uses a variety of commercial applications to generate invoices and keep track of his receivables. He uses two products from Shana Corporation for generating forms—Informed Designer to design forms, and Informed Manager to enter data—and the rest is done by AppleScript.

A script looks up all invoices, selects them according to certain criteria, lists them in the Scriptable Text Editor of Apple Script, and prints out the results. Smaller scripts are used to get customer balances, add details of a sale, pay a bill, and keep track of office timeslips.

NASA/Ames Research Center At NASA/Ames, federal regulations require tremendous amounts of paperwork, and extensive and precise approval processes; improperly filled-out forms cause delays in receiving materials that have been ordered. NASA/Ames generates as many as 20,000 purchase requests per year—of which half are routinely returned for some reason.

Mike Compton of NASA/Ames uses AppleScript to generate automated forms and review procedures. Building on a NASA internal expert tool called CLIPS, Compton used off-the-shelf products, such as 4th Dimension from ACIUS Inc., Informed Manager from Shana, and the Common LISP programming language, and created a validation system that allows a user to know immediately if a given form is not filled out properly. Once the form is completed, it is compared with a database for the corresponding authorizers' information in CLIPS. Required signatures are then added to the form. The new process results in considerable time savings and far fewer returns of purchase requests. ♣

because it permits the continued use of legacy systems. Client-server systems can be widely distributed, and can include diverse, multiple platforms while providing a unified access to data. Apple has created a model for client-server computing with VITAL (Virtually Integrated Technical Architecture Lifestyle), Apple's framework for integrating the desktop into the enterprise environment (see the front-page article in the July 1992 issue of *Apple Direct)*.

Although client-server tools are widely used in large enterprises, that is not the only audience. In fact, small businesses and education are increasingly finding ways to take advantage of these tools to speed up development and reduce costs. For examples of three organizations that are using client-server computing, see the sidebar "Client-Server Development in Action" on the previous page.

Whether you are a commercial developer, an in-house developer, or even a small business, what Apple is suggesting—in fact, the word *urging* is not too strong—is that you carefully consider the benefits of the client-server tools that are currently available.

From Legacy Systems to Personal Computers

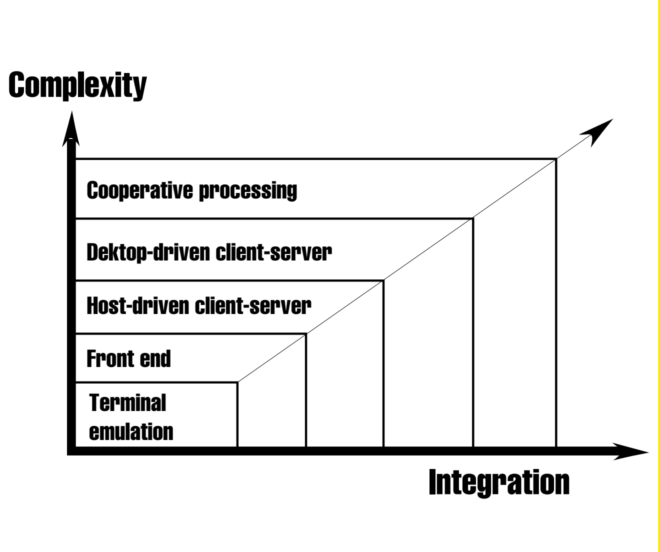
To understand the client-server model and how it became the prevalent computing model that it is in large enterprises today, it helps to know the history of software development in mixed environments. See "Greater integration leads to complexity" on the this page for a graphic representation of this history.

The traditional mainframe of the 1970s provided user access to data through terminal emulators, simple devices that were completely controlled by the host computer. While cost effective, this model provided no control to users, who were frustrated by not being able to access the data they wanted, and by the arcane functions and commands of a mainframe system.

This user frustration helped lead to the phenomenal rise in the use of personal computers in the 1980s, which put the user in charge for the first time. Over the following decade, personal computers grew in computing speed and were connected over increasingly powerful local area networks (LANs). But personal computers, even connected to a server over a LAN, simply cannot handle the computing demands that a mainframe system can; in addition, personal computers have remained to some degree incompatible with other computers. Most important, many companies simply could not afford the cost of abandoning the legacy systems that they had in place.

Thus over the last ten years, we have seen the marriage of mainframes and personal computers grow increasingly complex, and users are again faced with frustrating compatibility problems. Systems today are diverse, with wide varieties of semicompatible operating systems, network protocols, and software applications that are difficult to maintain and unwieldy to use. Users of such systems tend to have limited access to data, as well as a variety of user interfaces to cope with and master. As a result, a demand has risen for a comprehensive solution that permits legacy systems and personal computers to exist in harmony.

The client-server model has become that solution, offering the best of all systems—powerful data storage and computation on the mainframe, speed and ease of access over distributed networks, personal control on the personal computer, and consistency of interface over the entire system.



Greater integration leads to increased complexity.

The Client-Server Enterprise Model

Because the term *client-server computing* covers a lot of ground, this article divides client-server computing into four categories, according to where application processing takes place and where system control lies. These categories are

- distributed presentation (terminal emulation and front end)
- host-driven client-server computing
- desktop-driven client-server computing
- cooperative processing computing

Moving from distributed presentation to cooperative processing computing increases both complexity and integration of the entire system.

The traditional terminal emulation and front-end models leave all the computing power on the host. The front end handles only the data presentation function through the personal computer's graphical user interface.

In the host-driven environment, the application on the mainframe or mini computer is modified to communicate with the desktop, but the control remains on the host.

The desktop-driven model of client-server computing transfers more power to the desktop by placing a larger portion of the application on the personal computer. The presentation software resides on the client (the personal computer), and the data management module resides on the server (the host). Within this model, there are two varieties:

- the remote data access model, in which the application resides on the client, and the database resides on the server
- the distributed function model, in which the application is distributed between client and server, and messages are passed between them

In the desktop driven model, the host computer essentially functions as an information server. The application on the personal computer accesses services on the host, such as data access, transaction processing, and authentication and authorization.

The most flexible and dynamic model is that of cooperative processing, in which the host and the personal computers function as peers, communicating through messages that are passed back and forth according to a set of priorities. In cooperative processing, the integration of all systems into a unified set of peers allows for dramatic improvement in processing quality. Developers can create distributed applications that are customizable and adaptable to a wide variety of user needs and system configurations.

Client-Server Development

The client-server software development process tends to have three distinct characteristics: It allows for user-centered design, involves concurrent development, and uses object-oriented programming.

User-Centered Design. Unlike traditional host-centered software design, user-centered design creates applications that can be quickly and easily changed, while retaining the capability of accessing the host system. The user interface is friendly, and access to a variety of hosts is transparent to the user—one interface provides access to all the data that the user needs.

Concurrent Development. Concurrent (also known as *team*) development lets the developer work with modules of software, allowing for quick modifications to accommodate the users' needs. Unlike the traditional "waterfall" model of sequential development, the concurrent development model is iterative, whereby a given code module can be used and refined as needed, independent of the other code modules. If a new host computer must be accessed for its database, certain portions of the application code can be rewritten and deployed while the system continues to operate. (See the illustration on page 26.)

With concurrent development, programmers work with prototypes of each module, refining and improving the user interface or data access as needed. Over time, the system grows in efficiency and usability, but it can also be brought on line quickly. Businesses today cannot afford to wait years while a complex system is developed and refined. They prefer to get "up and running" with a working prototype that they can refine with user feedback as needs require and time allows.

This development model is also referred to as the "life cycle" of the application development process. Today's client-server tools are able to address all the stages of an application's life cycle, from CASE (Computer Aided Software Engineering) products that assist with analysis

Client-Server Tools Partners

Apple has already made two announcements of its client-server tools partners, an initial one on November 9, 1992, when nine ISVs announced their participation, and a second on April 6, 1993, when seven additional ISVs joined the partnership. Further announcements are expected this fall, when more companies will join as client-server tools partners. These companies are added to the list of Apple partners, including IBM, Digital Equipment, Oracle, and Symantec, all of whom provide sophisticated software development tools for the Macintosh computer platform.

Apple encourages you to contact these partners to see if you can work together to enhance the client-server capabilities of your products.

Last November, the following ISVs announced their client-server tools for the Macintosh platform.

Brio Technology DataEdit is a forms-based data entry system that provides Macintosh computer-based update capabilities for SQL databases in true client-server environments, making database front-end forms easy to build. Contact Gail Snyder McClain at (415) 961-4110.

Component Software Component Workshop is a dynamic object-oriented application development system based on C++ that supports cross-platform development to Windows and Apple events; it is used for developing, testing and debugging event-driven desktop applications running in a client-server architecture. Contact Susan Rutgerson at (617) 862-9700.

Digitalk Smalltalk/V Mac, an object-oriented system that is fully integrated into the Macintosh user interface, provides high-level support for native Macintosh computer operating system features such as QuickTime, QuickDraw GX, AppleScript, AOCE, DAM, DAL and Apple Events. With Smalltalk, developers can create applications with a single code base to run on Macintosh, DOS, Windows, or OS/2 platforms. Contact Barbara Noparstak at (310) 645-1082.

Forté Software Forté supports VITAL in the areas of desktop integration, data capture, and data access, allowing developers to build mission-critical applications. Forté simplifies the development of distributed applications through partitioned applications, open integration, GUIs, reliability, and performance. Contact Ed Horst at (510) 869- 3400.

Micro Focus Micro Focus has developed versions of the Micro Focus COBOL compiler as well as tools for Macintosh computers, including Micro Focus COBOL and Micro Focus Toolbox. Micro Focus COBOL applications are portable across UNIX, DOS, OS/2, and other proprietary operating systems and include tools for the creation, deployment, and support of cross-platform, client-server applications. Contact Eveline Kowtko at (415) 856-4161.

Mitem MitemView allows developers to create robust client-server applications for the Macintosh computer with mainframe-based legacy systems. MitemView ensures that an organization's highest levels of security are maintained and allows management of multiple, simultaneous host-server sessions, without any host code modification. Contact Barry Margerum at (408) 559-8801.

Powersoft PowerBuilder 2.0, an object-oriented, graphic, client-server development environment, is designed to build large-scale commercial and government applications. Powersoft is available on Macintosh and Windows platforms. Contact Jennifer Malloy or Rachael Stockton at (617) 229-2200.

and design, through the stages of prototyping and product creation, all the way to the maintenance stage, which is where many information systems departments spend most of their time and resources.

Object-Oriented Programming. In order to work efficiently with code modules, client-server developers usually work in object-oriented programming environments. Because object-oriented languages assign defined behaviors to objects, applications can be easily modified, and code portions can be easily reused in other programs.

In object-oriented programming, the developer creates a model for the work to be done by the application. Because objects can be easily reused by other systems, it becomes possible to place an object in the process flow of tasks to be accomplished by the application. Because such an object is self contained, it brings its functionality to the application transparently.

The features of user-centered design, concurrent development and object-oriented programming are now available in a variety of client-server development tools, not just for large mainframe environments, but also for a wide variety of developer needs.

A Number of Tools Categories

Client-server tools fall into a variety of categories. Some are analysis and design tools, such as Deft from Sybase—a set of CASE tools that can build entity-relationship, process, and data-flow diagrams using point-and-click commands.

There are application development environments, such as Apple's MPW, and Component Workshop from Component Software, which is completely object-oriented and maintains "live" objects that can be run at any time. Component Workshop is also an application framework (as is MacApp); these are C++ environments that go a step further by providing ready-made code modules that directly access Macintosh system routines, allowing the programmer to assemble applications quickly.

Other client-server tools are used to build interfaces or create front ends, such as "Both." from Connectivité, which allows developers to create Macintosh front ends for IBM mainframe applications, without modifying the host applications. MitemView from Mitem Corporation can be used for creating applications that access a variety of different hosts over dissimilar networking protocols.

Software AG Software AG's products for Macintosh computer-based client application development, server connectivity, and request distribution include Natural and Net-Work for Macintosh and Entire Broker. Combined with Apple's VITAL strategy, Entire Broker is designed to ease the complexity of enterprise networking and rightsizing. Contact Jen Norman at (703) 391-6727.

TGS Systems Prograph 2.5 is an object-oriented visual programming environment for the Macintosh computer. Prograph extension products allow support for DAM and DAL, and Apple's Interapplication Communication as well as Apple's Enterprise Toolbox. Contact Mark Szpakowski at (902) 455-4446.

Here are the ISVs who joined Apple's client-server tools initiative in April, and the products they announced:

Andyne Computing Andyne GQL (Graphical Query Language) is a fully customizable family of ad hoc data access and reporting applications that provide a multiplatform (Macintosh, Windows, UNIX Motif) desktop solution for enterprise-wide, end-user computing. Contact Scott Rankine at (613) 548-7801.

Blyth Software, Inc. OMNIS Seven enables application developers, such as MIS organizations and custom software development organizations, to build cross-platform, mission-critical applications capable of running on multivendor networks and accessing SQL and non-SQL databases. It supports client-server application development for Macintosh and Windows desktop systems. Contact John Mann at (415) 571-0222.

Connectivité Corp. "Both." 3.0 enables IS developers to retain their client applications—logic and presentation—while, behind the scenes, migrating incrementally to different data sources (such as SQL and LU6.2). "Both." is designed for rapid development, enabling the rapid creation of new, easy-to-use production-quality cooperative applications. Contact Lisa Grey at (914) 631-5365.

Inference Corp. ART*Enterprise features point-and-click generation of GUI, database mapping and application objects, multimedia, multi-developer support, unstructured information retrieval and rules, allowing creation of applications that link multiple heterogeneous databases and are portable across Macintosh, Windows, NT, OS/2, UNIX and MVS. Contact Chuck Williams at (310) 322-0200.

Sybase, Inc. Deft is a CASE solution designed specifically for RDBMS engineers and provides Data Flow, Entity-Relationship and Program Structure modeling. Users use it to design data-entry forms and report prototypes easily. Deft allows several engineers to work on the same design simultaneously. Deft generates SQL and forms from completed Deft designs and allows host databases to be reverse-engineered into Deft diagrams and dictionaries. Contact Linda Adreveno at (510) 596-3500.

Uniface Corp. UNIFACE offers a unique 3-schema architecture that allows developers to build, deploy, and maintain applications that are platform and database independent. The UNIFACE 4GL development environment for the Macintosh enables software developers to build enterprise client-server applications that transparently perform access to heterogeneous data sources across multiple networks on multiple servers. Contact Hal Steger at (510) 748-6145.

Visix Software Inc. The Galaxy Application Environment is a multi-platform development environment designed for building large-scale, distributed, graphical applications. With Galaxy's cross-platform API and integrated tool set, applications can run enterprise-wide on a broad range of desktop and server platforms, networks and graphical interfaces. Contact Sheri Winter at (703) 758-8230. ♣

A wide variety of client-server tools function as integral data management systems, which allow a customer or systems integrator to create custom databases that extract information from larger host systems, especially SQL databases, or build extraction tools that query the host system for key information and then present the information in a specially designed user interface. These tools include GQL (Graphical Query Language) from Andyne Computing, Brio Technology's DataPrism, and Omnis Seven Version 2 from Blythe Software.

Oracle Corporation provides a variety of tools for accessing SQL databases, developing applications, and porting Macintosh-built applications to and from many other platforms.

An integral part of the client-server model is the function of "middleware," which is the glue that holds disparate components together. Developers can access middleware with products such as Apple's Data Access Language Developer's Toolkit, which is a set of software components that provides access to host databases through application programming interfaces (APIs) that use standard SQL statements—even for the programmer who doesn't know the host computer's programming environment.

All of these tools are available now on Macintosh and either Microsoft Windows or Motif platforms—and many are available on all three. See "Client-Server Tools Partners" on page 24 for more information on Apple's client-server partners and their products.

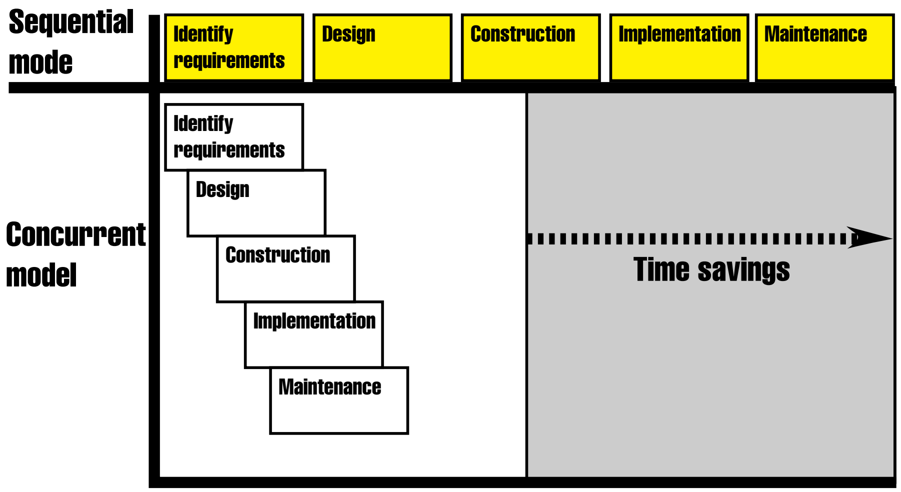
And of course, there's AppleScript, which defies categorization, but certainly is an integral part of many developers' client-server development efforts.

While some client-server developers use a single development environment, most use a variety of tools. The power of the client-server model is that different tools can be used for each phase of the development process. Using one set of tools to build a database and interface does not mean that you must permanently rely on only those tools; you can use other development tools to incorporate new hosts and servers, while maintaining a consistent user interface for data access.

What is important is that every stage of the life cycle is covered by the tools that are now available on the Macintosh platform. From analysis and design to prototyping, development, and maintenance, a variety of tools let you get the job done.

System Components Are There

Apple sees its role in the development process as provider of many of the essential components, including DAL, AppleScript, Apple events, and AOCE. In addition, Apple is positioning these components across platforms—they will be available for Microsoft Windows, UNIX, and MS-DOS systems—so that developers can write code that is usable on (or easily portable to) multiple platforms.



Concurrent vs. sequential development.

To facilitate cross-platform functionality, Apple is using (and recommends that you use) object-oriented programming languages such as C++ as much as possible. Because object-oriented languages provide features such as inheritance and encapsulation, you can write code to produce the best results rather than conform to the conventions of a particular operating system. With object-oriented programs, you can place the emphasis on application design, which you can then port to many systems while maintaining consistency.

Apple's core technologies (including AOCE and AppleScript) and the opportunity for cross-platform development now make true collaboration possible—where computers and applications function as peers, communicating through messaging, regardless of where the data or the various components of an application are stored. This type of computing is highly flexible and efficient and yields tremendous benefits for both users and systems administrators.

The client-server model for computing offers ways to dramatically reduce application development time, expand the power of existing and standalone applications, and, for commercial software developers, increase the market base for your products.

For more information, including a listing of many currently available client-server development tools, contact Apple's Client-Server Tools Evangelist David Berman (AppleLink BERMAN.D). Also ask about the *Client/Server Development Guide,* published by Apple Computer. ♣

David Gleason (AppleLink: DAVIDOVICH) is a regular contributor to Apple Directions.

Business & Marketing

Inside This Section

Thirty-five Macintosh Advantages Now Available

"Why buy a Macintosh? The short answer is this: People prefer Macintosh, are more productive with Macintosh, and are more satisfied with Macintosh. At Apple, we believe there are a number of concrete, specific, and, in many cases, proven reasons. In fact, we can think of at least 35."

So begins *Thirty-five Macintosh Advantages,* a new booklet prepared by Apple Computer, Inc., that can help you market your Macintosh products. Its 55 pages detail the 35 reasons Apple thinks Macintosh computers are better than IBM-compatible PCs running Windows, many of them backed by the findings of top independent research firms.

The reasons are divided into five areas: ease of use, applications, networking and connectivity, growth without disruption, value.

The booklet is full of facts, research data, analysis, and opinions you might find helpful when you put together your next product package, brochure, or marketing campaign. You may also want to share it directly with customers.

To obtain copies of the booklet, contact StartingLine at (800) 825-2145 and ask for item number L0440LL/A. ♣

Market Research Monthly

Exclusive: Apple System 7 U.S. Installed Base Numbers

In bringing new products to market, or releasing new versions of existing applications, you must make the choice between having your products offer dual support for System 7 and System 6 and requiring System 7 as the base operating system.

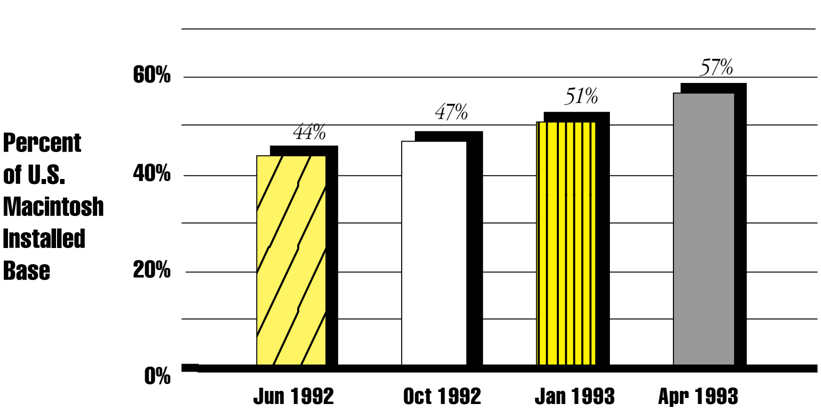
To help you make an informed choice, this month's Market Research Monthly features the first release of Apple Computer, Inc.'s proprietary System 7 penetration data, as we promised at the Worldwide Developers Conference in May 1993. The information provided in this article is based on the most recent quarterly Macintosh System Software Marketing survey of the U.S. Macintosh installed customer base, completed in April 1993 and updated in July. Apple performed the study by using a random sampling of U.S. Macintosh users who purchased computers between January 1986 (after the introduction of the Macintosh Plus) and December 1992.

System 7 celebrated its second anniversary in May 1993. In two years on the market, System 7 has proven a strong business success. Many customers and developers alike have adopted System 7 as their standard. Over 6,000 applications run on System 7, with hundreds taking advantage of specific System 7 features such as Apple events and TrueType support. A growing number of applications now require System 7.

Overview

Overall, of U.S. users (numbering approximately 6.5 million, according to a variety of industry sources), 57 percent are currently using or are in the process of acquiring System 7. Of those who previously used the Macintosh with System 6, 29 percent have upgraded to System 7. How satisfied are these System 7 users? Of those upgrading from System 6 to System 7, 84 percent

System 7 Penetration Rate in the United States, 1992–1993



report a high degree of satisfaction with the new system software.

Additionally, since the introduction of System 7.1 in October of 1992, 12 percent of the U.S. Macintosh installed base now uses System 7.1. See the pie chart, “Who’s Using What Macintosh System Software?” for a breakdown of current usage of System 6, System 7, and System 7.1 among U.S. Macintosh users. (Throughout this article, total System 7 figures include System 7.1 users, unless specifically noted.)

System 7 Use by Macintosh Model

The graph “System 7 Acceptance in U.S. by Macintosh Model” shows usage of System 7 among the U.S. Macintosh installed base by the kind of Macintosh computer. As the graph shows, the System 7 acceptance rate is over 80 percent among users of “mid-range” and “high-end” Macintosh computers. Thirty percent of “low-end/compact” Macintosh users have adopted System 7. (See “System 6 Users” for more information on this category.)

System 7 User Characteristics

Macintosh customers using System 7 are among the most active users, making greater use of peripherals—modems, scanners and CD-ROM drives—than do non–System 7 users.

Users who have upgraded to System 7 are usually experienced Macintosh users. The majority have been using a Macintosh computer for more than four years. Those who have upgraded use a computer in the Macintosh II family more than any other models.

System 6 Users

What about customers still using System 6? Eighty-six percent of them are aware of System 7, but see little reason to make the transition. Anecdotal data suggests that they are satisfied with their current system software and/or feel they don’t use their computer that much to warrant upgrading.

Those who have not adopted System 7 are generally less sophisticated Macintosh users. They are more likely to use a lower-end or compact Macintosh (Macintosh Plus, SE, SE/30, LC, or Classic). They are also less likely than System 7 users to use peripherals, such as a CD-ROM drive, scanner, or modem.

Developers’ Choice

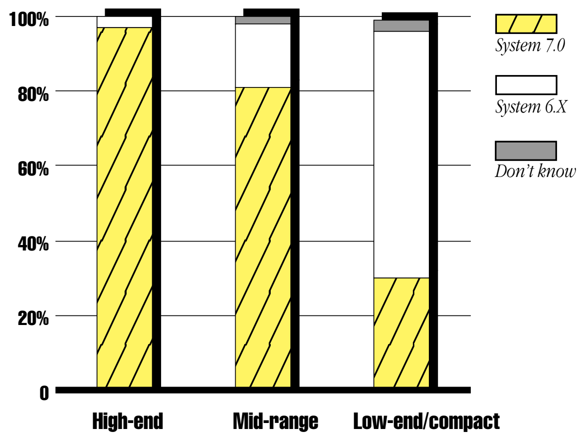
In trying to decide whether your products should offer dual support for System 7 and System 6 or require the use of System 7, you should consider who their target customer is. For example, an entertainment application targeted to the home market should probably support both System 6 and System 7, while a business application, such as a presentation graphics program, may require System 7.

Apple has advocated full support for System 7 since its introduction. Apple shifted its business entirely to System 7 within six months of its launch, shipping it on virtually all Macintosh models. Since the release of version 7.1 in October 1992, all new Macintosh computers worldwide ship with System 7.1.

You’ll want keep in mind that System 7 will remain the standard Macintosh operating system for the future, shipping with the next-generation PowerPC processor–based Macintosh computers that will begin to ship in the first half of 1994. Future releases of System 7 will offer preemptive multitasking, memory protection, and sophisticated system extensions, such as AppleScript, OpenDoc, QuickDraw GX, and AOCE.

Apple Directions will update this U.S. report with worldwide data when it’s available. We’ll also continue to bring you timely market research data, generated by both Apple and leading industry analysts. Send an AppleLink message to A.DIRECTIONS and let us know the data you’d like to see in future issues.♣

System 7 Acceptance in U.S. by Macintosh Model

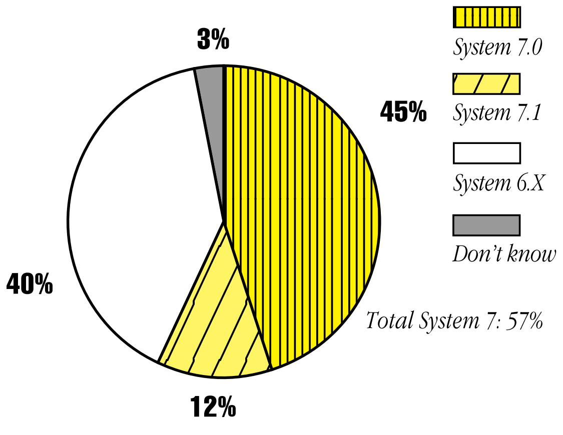


High-end Macintosh:
Macintosh IIfx; Macintosh Quadra 700, 800, 900, 950.

Mid-range Macintosh:
Macintosh LC, LC II, LC III; Macintosh II, IIx, IIsi, IIcx, IIci; Macintosh Portable; Macintosh Performa 200, 400, 600; Macintosh PowerBook 140, 145, 160, 165c, 170, 180; Macintosh PowerBook Duo 210, 230 (with and without docking stations); Macintosh Centris 610, 650.

Low-end/compact Macintosh:
Macintosh Plus, SE, SE/30; Macintosh Classic, Classic II, Color Classic; Macintosh PowerBook 100.

Who’s Using What Macintosh System Software?



100% = U.S. Macintosh installed base

Marketing Feature

If We're So Smart, How Come We're So Dumb?

Positioning Your Message Using the Buyer's Perspective—Instead of Your Own

By Geoff Moore, Geoffrey Moore Consulting

The other day I was reading the September 1992 issue of *Apple Direct,* specifically, "The Ten Commandments of Product Packaging: How to Create Packages that Sell," by Signe Ostby of Merrin Information Services. Each of Signe's commandments is, well, common sense (sort of like the original ten, come to think of it). The problem is, we often don't obey them (sort of like the original ten, come to think of it). Why not?

For example, let's take Signe's opening premise, as clear a piece of common sense as you would ever want to hear: "The package's primary purpose is to sell the product inside it."

This statement is so obvious—particularly to people as smart as we are—that it wouldn't be worth making, except for the fact that so many packages simply *don't* sell the products inside them. If we're so smart, how come we're so dumb?

Speaker- vs. Audience-Centered Communication

To understand this, consider that all "speech-acts," as some linguists like to call them, can be placed on a continuum between speaker and audience, depending on whose needs they are most focused on fulfilling. A very audience-centered message might be, "Watch out!"—especially if audience members were standing under a falling rock, for example.

But what if the audience were deaf, and the speaker knew this fact yet still yelled out, "Watch out!" Now what's going on? The speaker's speech-act is now a form of self-expression, a release of stress, if you will, something he or she just "had to do" even though the verbal warning could not communicate to the audience. The same thing goes for yelling "Ow!" when you stub your toe, whether or not anyone is in the room. Or singing in the shower. Or doodling in a meeting. These are all speaker-centered forms of communication.

These are the extremes of the speaker-audience continuum. As we move more toward the middle, the question becomes more problematic. If I write a software application and then tell you, "This is a really cool program," whose needs am I serving? Clearly mine, for starters—I can't wait to tell you, in fact; I'm just bursting with self-expression. But, if it really is a cool program, then I'm also serving your needs by letting you know that.

Now that works fine if we're friends. But what if you don't know me? How much leeway do I have for "self-expression"? How much do you want to hear from me? According to the research, you don't want to hear very much—indeed, you want to hear far less than I want to tell. And that's the problem we face in formulating the communications vehicles that carry our positioning messages to prospective buyers.

We know that we're supposed to be audience-centered, but when you look at our output, you see that all too often we've been unable to overcome the speaker-centered pressures that act upon us. To paraphrase the title of a popular self-help book: *Our Packages, Our Selves.* (This is actually but one title in a series, others of which include *Our Logos, Our Selves; Our Advertisements, Our Selves;* and the most popular of all, *Our Product Names, Our Selves.*)

Unfortunately, "our selves" don't sell. Selling is about the buyer, not the seller. It is the buyer's self that sells, not *ours.* We need to resonate with the buyer's self, but we can't until we banish our own. So to be painfully clear about this, it is not our package, or logo, or advertisement, or product name—it is *theirs.* And the issue is not whether we like it, but whether *they* like it.

Before the end of this article, I'll suggest some practical procedures for integrating this insight into your everyday communication efforts; but before I do that, we need to fully explore the dangers that surround you.

Beware the Designer

First of all, it isn't just you that has the speaker-centered problem. It's all of us. Take an ad designer, for example—author of *My Design, My Self,* subtitled with an appropriate Shakespearean echo: "A CLIO! A CLIO! My client for a CLIO!" (A CLIO is a prestigous television advertising award.)

Designers have agendas just as we do. I witnessed a public manifestation of this syndrome recently at an industry conference, during a panel on package design. A principal of a highly respected design firm was reviewing packages (in pairs) and pointing out the virtues of one versus the defects of the other. A slide flashed on, depicting a quite presentable box on the left, whose product name I don't remember (that, by the way, is what detectives call a *clue*), and on the right was the ever-memorable, phenomenally schlocky package for My Personal MailList (MPL).

For those of you who are so Mac-centered as to have never seen this box at, say, the checkout stand of your local superstore, the front panel portrays a prettyish, thirtyish '50s-style woman (Betty Crocker hairdo, June Cleaver facial structure), hands to her face in a golly-gee-whiz expression of astonishment as hundreds of letters rush out of her keyboard (which has been transformed into a mailbox) and into the postal system. Straight out of *Popular Mechanics* thirty-odd years ago.

The speaker was quite rightly praising the aesthetics of the box on the left and bemoaning the same for the MPL package, when someone from the audience had the temerity to note that MPL was outpulling the other product in the marketplace at a ratio of about 100 to 1—to which the response from the podium was: *Click. Next slide.*

I believe this is what psychologists call *denial.*

The point is, since the MPL package violated even the simplest rules of aesthetics—and trust me, there's no doubt that it does—it violated the speaker's (a designer) value system, and thus *it could not possibly be good design.* The problem, of course, is that buyers didn't agree. They didn't have the "right" value system—so they ate it up. (Peasants!) In any event, this designer simply would not have it. No design like that would ever come out of her shop; it would ruin *her* reputation.

And the truth is, it might. But the problem is, this kind of design is what the audience in this particular market segment preferred. And if we're going to ask ourselves not to be speaker-centered in communications, we must be able to hold our design agencies to a similar standard (and not flog them when they deliver on it).

Now we need to bring another component of the communications problem into focus before we can move on to finding solutions.

The Fantasy/Reality Model

There is a second continuum in the realm of communications that cuts across the speaker-audience axis, and that is the one between fantasy and reality. In communications, this is a critical continuum to manage.

The role of fantasy in marketing is to communicate benefits that stimulate buyers' appetites. Customers expect some level of fantasy experience during the buying process. They want it. It's fun. It's why people browse in stores, even when they don't intend to buy anything.

At the same time, customers don't like to be fooled. They know all too much about "buyer's remorse" ("Why did I buy that?"), so they want to be able to do a reality check before they make a purchase. How much reality checking customers need depends on how risky they perceive the purchase to be. In business-to-business sales, the scales tip towards reality. In entertainment products, they tip toward fantasy. The point is that it's the audience's call.

Moreover, the types of fantasy- and reality-based information that go on a package or into an advertisement should likewise be the audience's call. And so now we have the entire "message focus" model (as illustrated below), to help us see where we go wrong and how we can correct it.

The first principle of the model is that communications should be driven by the right side of the diagram (audience), not by the left (speaker). There are two forces that drag your efforts back toward the left. The first is the need to "self-express," whether it be on the part of the developer, the communicators, or the designer. The second is the desire to use a package, ad, or logo as a status symbol among your peers. Neither of these forces is legitimate, but both have power over you.

The second principle is that you should correctly weight the relationship between fantasy and reality relative to the audience's purposes. In a business-to-business sale, where the audience weight is toward reality, speaker-based fantasies are the worst communications sins. These are most often committed by vice presidents of sales. Being charismatic types, they're accustomed to creating reality distortion fields during sales presentations—and then holding them in place by sheer force of personality. They want that same distortion field to permeate all of product communications.

Print media communications, however, simply don't have the force to pull this sort of thing off. Moreover, in a situation in which the audience has real-world worries, fantasy-based appeals are often resented. Consider, for example, a recent Computer Associates ad showing wild horses splashing along the edge of a lake and captioned "No Fences. No Boundaries. No Limits. That's the Freedom of CA90s." This is an exercise in promoting a speaker-based fantasy that MIS directors aren't likely to share. CA90s are enterprise-wide information systems that require compromises to implement. My feeling is that audience members aren't likely to be attracted to this ad, and some may even resent it. (Just my opinion, of course.)

On the other hand, in the same magazine Sony ran an ad for its optical disks, with the caption "Write Once. Read Many. Worry Never." This is also a speaker-based fantasy, but it will probably work with the audience for two reasons: There is much less at risk in buying an optical disk than an enterprise-wide system, and Sony has a reputation for quality, probably more than CA does for freedom.

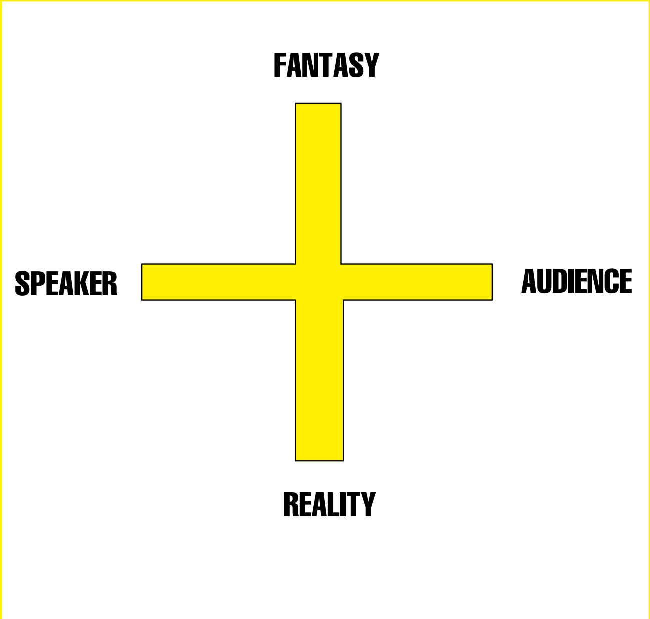
In most instances in business-to-business sales transactions, the vendor's reputation and credibility set the limits on how far the audience is willing to go along with such claims and therefore be charmed by marketing communications fantasy.

To put this in perspective, let's turn to a consumer packaged-goods type of transaction, where the audience's expectations of communications pleas are tilted more toward fantasy. Here the worst communications sin is to indulge in speaker-based "realities"; this sin is most often committed by vice presidents of engineering. To avoid this error, for example, you can put a fantasy on the front of the product package. However, you can also err simply by putting a speaker-based fantasy on the box instead of a customer-based one.

For example, recently the makers of the Aladdin game redesigned the package front, and sales shot up 30 percent. What happened? Well, the earlier box showed a movie poster image with Aladdin sweeping a harem maiden into a Rudolph Valentino embrace. The new box shows Aladdin all by himself leaping into the air and performing a kind of stylized Ninja kick. Now, which image do you think would have more appeal for a nine-to-twelve-year-old boy?

Aesthetics and Selling

Finally, let's return to the matter of aesthetics. The aesthetics of your communications pieces send value signals to an audience.



Message Focus

This creates a problem in high-tech marketing communications for several reasons. First, aesthetic values are inherent in all marketing communications, so they are present in all our transactions—regardless of whether we want them. Second, aesthetic values strongly correlate with perceived social status and class distinctions. This makes it very difficult for people, whether they're developers or designers, to opt for using audience-centered aesthetics, because they think peers will make personal judgments about them based on their communications pieces.

The thing that was great about the My Personal MailList example was that its creators saw how a schlocky aesthetic could convey an appealing and reassuring marketing message, and they didn't hesitate to use it. The fantasy on the front panel of the package is ludicrous, but for anyone who has to type in mailing lists, it's certainly appealing. And the fact that it's done so schlockily means that the customer is not really expected to take the fantasy seriously.

Furthermore, this style conveys yet another message—cheap (as in inexpensive, not low quality). And that's exactly what MPL is, selling at well under $50 a copy. This cheapness is also reassuring, because it implies that the product is not going to do very much (that is, it isn't complicated), and thus won't be too hard to learn or use. And, finally, the '50s-style imagery is so old-fashioned that it is reassuring. This is a product, it says, for people who are not crazy about high tech, but who have to use it anyway. All in all, a very effective image.

Another issue relating to aesthetics further complicates the problem. High-tech marketing communications is usually a joint effort between engineers, who often lack an aesthetic sense, and designers, who have an overdeveloped one. These designers tend to believe it is their role to defend aesthetics against the assaults of the Philistine engineers; and designers can become so militant that even when their aesthetics are inappropriate for the product's audience, there is no way they can "hear" this feedback.

Worse, often a product with a package design that makes a great aesthetic statement simply does not sell, but instead sits on the shelf inertly, an impressive icon, but not something that attracts customers. So, for a variety of reasons, the aesthetic statements embedded in our communications are often out of sync with our marketing strategy, and business professionals who sense this are feeling increasingly irate about being held hostage by the value judgments of the designing profession.

There Must Be Some Way Out of Here

So how do we proceed given that, for most of us, this whole business of graphics, packaging, communication, and aesthetics is not a strong suit? Here's my advice, formulated as seven admittedly California-ish beatitudes, in an attempt to add some extra balance to the admonishments of Merrin Information Services' ten commandments:

#1: Blessed are the well-informed, for they shall not seek to drive to Hawaii. Clip out Signe's packaging article or get a reprint from Apple. Every piece of advice it gives is sound.

#2: Blessed are they who know themselves, for they will not order pizzas for twenty, all with anchovies. Before starting work on any communications piece, such as packaging, form a clear image of your own personal style and that of any consultants or agency people you are working with. In the case of packaging, the easiest way to do this is to go to a superstore and simply point out to each other packaging that looks good to you. Or for ads, flip through some magazines together. Don't let this become a matter of right versus wrong; in aesthetics there really are no right or wrong answers, only matters of taste and class distinction. Instead, use it as an exercise in understanding each other's tendencies for self-expression.

#3: Blessed are they who are willing to know others, for they shall be able to distinguish windows from mirrors. Build a model of the audience, via a personal profile of an idealized target customer. Make sure this fictional person is clearly differentiated from everyone on the marketing communications team (unless you want to make one member of your team stand in for this person). Focus specifically on differentiating the target customer's values from your own.

#4: Blessed are they who are sensitive to the situation, for they will not throw to second base when there is a runner on third. Characterize the perceived risks and rewards of the buying situation from the target customer's point of view. Determine from this what kind of mixture of fantasy and reality he or she will desire. Capture that as a percentage ratio to be used as a test during communications/package evaluation sessions.

#5: Blessed are they who put things in their proper place, for they will be able to find the flashlight when the power goes out. In the case of packaging, allocate the front panel of a box to fantasy, the back to reality. The idea here is that the front hooks the customer into picking the box up and turning it over, and the back gives them permission to buy. (Like any general rule, this can be violated to good effect, but use it as a starting point and realize that when you do go against it, you are committing a violation.)

#6: Blessed are they who check themselves over carefully, for they shall not leave their credit cards in strange places. Make sure the fantasy is audience-centered and not speaker-centered. If it appeals to you strongly, make sure this is because you really do empathize with your customers, and not because you have substituted your fantasy for theirs.

#7: Blessed are they who can walk in another's shoes, for they shall not have to go barefoot on pavement on a hot day. Make sure the reality information on the back of a package is purchase-decision related. (The same applies to the body copy of your ads or brochures.) If it isn't something your customers need to know to help them buy, don't say it. It will only distract them from the information they do need to get.

If we go forward with these principles in our hearts, it shall come to pass that not only will we continue to be smart, but we will also, maybe for the first time, not be so dumb. ♣

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

How to Choose the Right Product Name

By SB Master,
Master-McNeil, Inc.

In a time when high-tech products are produced with the frequency of baby rabbits, developers must use everything at their disposal to distinguish their products from the rest of the pack. Choosing the right name for your product can be the element of distinction that determines whether customers give your product a try.

But finding an appropriate name can be a serious test of character for a marketing team. Recent revisions in trademark law and the overwhelming number of new products entering the market have made finding a suitable *and available* name more difficult than ever. However, by following a well-organized naming plan and allowing yourself enough time, choosing a product name can be rewarding—and even fun. This article will outline the basic principles of product naming and will define a process for helping you match your great new product with a truly stellar name.

What a Name Will Do for Your Product

Once you realize that a powerful name can play a key role in launching a product in the right direction, the energy required by a serious name development effort will be easier to justify and your efforts will be more rewarding. But how does having a good name affect a product? Our views are summarized below:

• *Good name, bad product.* A great name can help kill a bad product. A great name will encourage people to try a product, and they'll thereby find out sooner that the product is bad.

• *Bad name, bad product.* This one is obvious: Forget it; save your time and money.

• *Bad name, good product.* A great product can usually overcome a bad name, but a bad name can slow down trial and acceptance. Why burden your great product with anything less than a great name?

• *Good name, good product.* This is the best situation you can hope for, and is definitely worth the effort. We should all seek this powerful conjunction of product and name.

Characteristics of a Good Name

One of the most commonly asked questions in the naming business is what separates good names from bad ones. This isn't really the relevant question; what really matters is whether the name communicates the right messages to the right people. So, the crucial first step in creating a product name is determining exactly what you want your name to express, and to whom. These qualities will be embodied in your naming objectives, described later in this article (see "Step 2: Develop a List of Naming Objectives").

There are several basic categories of names from which to choose. (For information about these categories, see "Kinds of Names" on page 35.) But whatever kind of name you choose, it must meet all of the following criteria:

• *Good names are easy to pronounce.* An effective name rolls off the tongue. (Can you imagine having to provide a pronunciation guide in your product introduction literature? Some companies actually do that.) But even more important, the name must be difficult to mispronounce. This point can't be overstated. If you give customers an opportunity to mispronounce a product name, chances are they will. Or, even more fatally, they won't talk about your product at all because they fear the embarrassment of mispronouncing the name. When this happens, you stand to lose one of the most valuable, yet least expensive, communications media available to marketers: word of mouth. The pronunciation of your product name must therefore be totally without ambiguity.

• *Good names are memorable and appealing.* When it comes to choosing one product over another, all other things being equal, the product with a name that sticks in the customer's mind will probably be the stronger candidate. Ease of pronunciation can be evaluated objectively using linguistic principles, but memorability and appeal require subjective judgments.

Some names are definitely more memorable and appealing than others. These names are not only easy to pronounce, but they also tend to be fun to say, to be spelled conventionally, and to relate in some manner to the product category or benefits. A totally arbitrary name which has nothing to do with anything, and that is unpleasant to say, will tend to be unappealing and not memorable. But a name that strikes a responsive chord, or that manages to attach itself to an existing "hook" in the customer's mind, and slips easily off the tongue, will tend to be more memorable and appealing. Some good examples of memorable and appealing names are SimCity, FinePrint, QuickDraw, FreeHand, WriteNow, PostScript, Persuasion, and Timbuktu.

• *Good names are free and clear of legal problems.* You need to ensure that a name will be available for use in all your key markets. A name should be trademarkable, or else should be so descriptive that you don't need a trademark for it. (Names that are very descriptive often use words in common use in the language and often aren't trademarkable.) Get advice from a trademark attorney.

• *Good names should be effective worldwide.* To be effective worldwide means that beyond being legally available in all key markets, a name should be pronounceable, have no negative meanings, and have no embarrassing connotations in the countries, cultures, and languages in which it will be used.

You should consider this issue even if you have no immediate plans to market internationally, for two reasons. One, even if you intend to market only in your home country, there usually are large numbers of potential customers who speak languages other than your native tongue. Two, if your product succeeds, you may eventually want to market it outside your home country. It is far more cost effective (and better worth the time and effort) to build a worldwide identity for your product than to change its name country by country.

There are a variety of considerations regarding what makes a name appropriate in multiple markets. As Lauri Jones of Intracom, Inc. pointed out in an earlier *Apple Direct* article, most technical product companies keep a product's original name when changing markets. This hasn't always been true of more consumer-oriented marketers, but it's becoming more common. So, it's important to ensure that as you take your name from country to country, it does not take on negative connotations in other languages. A classic example is General Motors' attempt to market its Chevrolet Nova model in Latin America. *No va* in Spanish means *doesn't go.*

A cost-effective way to check the multiple language associations of your chosen name is to review it with native speakers of various languages within your company. Quick checks by electronic mail or phone with foreign associates also work. Be sure to talk with native speakers, not just someone who happened to take a few years of the language in college; you need to be aware of slang, archaic, regional, and street language associations as well as textbook or dictionary meanings. To be truly safe, you should consider enlisting the services of localization experts to evaluate your name in foreign markets.

Now you're ready to begin creating your name. A key consideration is when during product development to start the naming process. Set up a schedule working backward from the date the name is required; typically this will be the date of the initial product announcement, or the date you need to drop a name into product manuals or packaging. If you are a startup company, you may need the name much sooner: When talking to investors, your product concept will seem much more tangible if the product has a name.

How much time to allow for naming depends a lot on the size and culture of your company, how many people need to participate, and how many levels of approval are required. Generally, a twelve-week schedule should be adequate to take you through the naming process described in the following paragraphs.

Step 1: Organize a Naming Team

Setting up a naming team is the first—and often overlooked—step in creating an effective name. While a one- or two-person team might make the decision process easier, you'll eventually have to defend your choices to a larger group and you'll lose the time you saved. A team effort is better: It leads to increased participation in the naming process, and a more diverse range of opinions about what the name should convey. You'll end up with a happier staff and with more people having a pride of ownership in the name of their new product.

A naming team should include six to eight employees who represent a variety of viewpoints and job functions; the team should include at least one person who is likely to be a key participant in the eventual implementation of the name, such as someone from public relations, marketing communications, or advertising. Team members ideally should include a mix of mid- to high-level employees, all having some degree of familiarity with the product.

If there is someone in the company who is likely to hold widely divergent or controversial views and who has the potential to veto or slow down the process, don't put off the confrontation—include that person on your team. Believe me, this works better than isolating dissenters and then later trying to convince them of your wisdom. By having a variety of viewpoints and job functions represented on your naming team, you'll go a long way to getting the necessary buy-in from the company as a whole.

The naming team's job is to carry out the steps that follow. Team members will also play a role in explaining the project process and results to fellow employees.

Step 2: Develop a List of Naming Objectives

The team's next step is to establish a list of naming objectives. These should reflect the marketing objectives for the product, including such things as who the target audience is, how competitors are positioned, pricing, distribution plans, and relationships to existing company products.

Naming objectives define the ideas or characteristics the name should convey. In high-tech companies, objectives often encompass such concepts as innovation, efficiency, dependability, speed, or high quality. The naming team must establish a list of the most important goals specific to your name. What do your potential customers need to know to make them consider your product? What key words or concepts already exist in your product category—and do you want to leverage these or avoid them? Will the name need to stretch over any other products? These key thoughts should be embodied in the naming objectives.

Why is it important to conduct this objective-setting exercise? Because once you have agreed on a list of objectives, you can use it to evaluate the candidate names.

For objectives to be useful, they must be specific, discrete, and unemotional. Try to keep the list short, and establish a priority order. For example, a set of objectives for a new family of software utilities could be

- to suggest an "umbrella," a family of products
- to suggest utilities, tools, assistants, aides—not complete solutions
- to suggest increased efficiency, productivity, and speed

Note that these objectives are dry, rational, and unemotional. They do not overlap or contradict each other. They lend themselves to the kind of name evaluation you'll have to do later: Does the name suggest a family of products—yes or no? Does it suggest a utility or a complete solution? And so on.

In contrast, below is an example of a less useful set of objectives. They are confusing, unspecific, and contradictory; there are too many of them. They don't lead the team in any particular creative direction and they will give you little help when evaluating names:

- to be catchy, something that grabs you, or funny
- to suggest innovation, the leading edge
- to suggest the mainstream, a safe choice, reliability
- to suggest leadership, importance
- to suggest friendliness, approachability, a helpful quality

I think you'll agree that these objectives are vague, inconsistent, and less useful than those in the first list.

Step 3: Develop a List of Naming Criteria

Naming criteria function as technical specs for the name. They dictate how your name will express the ideas stated in your objectives. For example, here is a set of criteria to accompany the objectives for our family of software utilities. The name must

- fit in with other products in the existing product line
- contain no more than two syllables (must be short because it may be combined with company name and individual product identifiers)
- avoid use of terms used by competitors: *power, mate, multi, set*
- be pronounceable and legally available in the countries in which you plan to market the product (list them specifically)

As with objectives, criteria should be simple, specific, uncontradictory, and few in number.

Step 4: Initiate the Creative Work

Once team members thoroughly understand and agree to the objectives and criteria, you're ready to begin the process of creating the name itself. Don't underestimate the effort needed at this point in the process; while it can be fun, developing names is more difficult than it appears.

To give team members confidence and to help focus their creative efforts, provide the team with a synopsis of competitive naming practices, and discuss whether competitive names make sense or represent directions to avoid. Also provide the team with lists of word parts from English, Latin, Greek, or other relevant languages to give them raw materials for expressing your objectives in less obvious ways. These word-root studies are excellent springboards from which team members can explore new directions for names.

To construct the list of roots, start with key words from your objectives list and explore both where those words come from and how other languages express those ideas. The resulting names will not all be easy to understand (not all of the roots have easy connections to English), but many will make unique and interesting name alternatives for your product. For example, from the software utilities objectives, a root study of the word *speed* would include such word parts as *agilis* (Latin for *quick*), *celer* (Latin for *swift*), *presto* (Italian for *quick*), and *tachos* (Greek for *speed*). These could all be used to construct new names.

Each team member will interpret the word parts from a different point of view, yielding an expansive list of coined but intrinsically meaningful names that are less likely to be already in use.

Also, construct a list of any names that the team and other company employees have previously suggested. Discuss what about those names made them interesting (tone, meaning, balance, character), and try to create more names in those same categories. Finally, provide your team with dictionaries, thesauruses, directories of existing products, competitive promotional literature, and anything else you think might help.

Then, turn the team lose. Ask each person to develop a long list of names. Encourage team members to continually refer back to the original objectives and criteria, to help them stay on target.

Step 5: Review Creative Work Together, Then Do More

After the team has had time to work independently, you should get together for a team meeting. How soon? Probably after team members have each devoted at least eight hours of focused effort to name development (ideally spread over three to five days). The objective for this meeting (which everyone should know before you start) is to review the creative work with an eye toward establishing preferred naming directions, highlighting interesting ideas for further exploration, and uncovering any new revelations.

Open the meeting by reviewing your naming objectives and criteria, to ensure that everyone remembers the goals they've agreed upon. Then explore the creative work, highlighting the positive and avoiding any judgment or ridicule. Create a feeling of progress, of zeroing in on the ideal name. Agree on which ideas are promising and worthy of further attention, and which ideas have been adequately explored.

Then, send the team away for a second round of creative development. Team members should explore the preferred naming directions, exploit new information, and probe more deeply into the creative possibilities suggested by the objectives. This will lead to an entirely new list of names.

It is often this second, more focused and educated push, that results in the most interesting name alternatives. Continue the process of creative exploration-team meeting-creative exploration until you think the objectives have been exhaustively explored. You could easily end up with a master list of a thousand or more name candidates.

Step 6: Narrow the List

Each subsequent step from here is a gradual convergence on the best names from your master list, carrying forward only the best alternatives. Review the master list and select the most promising candidates. These are the names that, in the team's judgment, best express the objectives and meet the criteria that were agreed upon earlier. Though it may take a few fist fights (or some bribery), the team should agree on a smaller list of 40 to 50 names before taking the next step: a preliminary search to determine if the names are already being used in the marketplace.

Step 7: Preliminary Name Searches

The preliminary search for market presence can be done in several ways. In the United States, the simplest and, in general, most effective method is to use an online database such as TrademarkScan that lists current, pending, and canceled U.S. trademarks.

There are a few key pitfalls to avoid when conducting these searches. One potential problem is spelling variations. Trademark law is designed to prevent confusion among products in the marketplace, so two names that are spelled differently but pronounced similarly may be rejected. You'll need to check similar spellings and plural versions of your name, as well as exact matches. A name such as *LaserWriter,* for example, could be spelled *LazerWriter, LaserRighter,* or *Laser Writer.* You need to think of and check all of these.

You also should consider multiple trademark classes. A trademark may be registered within one or more classes, each class representing different product types. You need to search not only the class into which your product falls (electronic goods and services, for example), but other potentially overlapping classes such as entertainment, telecommunications, and printed matter. To do this, you'll need to consider the product categories into which the name might eventually grow, not just where it is today. One example is the software that became a television program, "Where in the World Is Carmen Sandiego?" The advent of multimedia products has made it even more important to check name availability in multiple product categories.

A third pitfall is prematurely discarding name candidates. While it's true that the previous existence of a name in the market can cause problems for your candidate name, information gathered through preliminary searches is not always accurate. Many products that show up as potential conflicts may have been discontinued long ago. Their parent companies may have disappeared. The product may have never been introduced. Some names may be for products different enough from yours to avoid any conflict. Or names may be owned by very small or nearly defunct companies, who might not object to you using their names, particularly in exchange for a small fee. Therefore, it's important when examining the results of your searches to keep a close eye on all the information provided, and investigate the companies listed as using your name candidates. Many excellent candidates will otherwise be tossed out.

Once you've put a list of names through the preliminary search process, you can learn even more about their potential availability by checking some other readily available databases. The United Kingdom version of TrademarkScan, which tends to turn up major European trademark users, is now available online. Other useful methods include literature scans, which search for articles in industry journals that might mention your candidate names; the Electronic Yellow Pages, which turns up companies of all sizes using your candidate names as company names; and the Companies And Their Brands database, which shows products whose names owners may not have bothered to trademark.

Step 8: Select Your Finalists

You now must decide which of the names that have passed scrutiny are the most worthy to pursue. By this stage, all the remaining names will have passed many tests. But you'll probably still have too many names. How do you decide which ones to take forward?

Go back to the objectives and criteria one last time; consider the names' longevity and extension possibilities, and their potential international acceptance. Some names may express one particular objective well but leave others somewhat neglected. If so, your team must decide which objectives carry more weight and which tradeoffs make the most sense. Through this process, select a group of five to seven names, any one of which would be a good solution.

Step 9: Have an Attorney Conduct a Full Trademark Search

A full search for a name's legal availability should be turned over to a trademark attorney. The attorney will need to do a full U.S. search, as well as international trademark searches in key markets.

Review the results with your attorney. If only one of the names submitted is available, you have found your name. If more than one is available, your last step is slightly more difficult, but still straightforward: Choose one.

If you were unfortunate enough to lose all of your final candidates to existing trademarks, don't panic. Review your most promising name list (or the original master list, or even do a third round of creative work) and select a new batch of names that meet the objectives and criteria. Then submit these names to a preliminary search and continue the process through doing a full search.

Build In-House Support for Your Name

As if finding a legally available and appropriate name weren't difficult enough, you'll now need to convince your associates that you have found the best name. This task will be easier if your team was constituted as I recommended earlier; members representing various departments and interests will probably have kept their colleagues somewhat informed as to where the naming process was going, so its outcome won't be a complete surprise.

Depending on your organization, who needs to be informed, and when, will vary. The first task is to get approval for your selected name at whatever level your organization dictates (such as the vice president of marketing, for example). A formal presentation (though the definition of *formal* varies) is recommended. In your presentation, the naming team should review the process it followed; it's important that people understand and appreciate that your name recommendation is not being made lightly. Show the steps you took, the people you talked to, the competitive practices reviewed. Spend time on the objectives and criteria—why you chose them, which ones you left out. Review any international or customer input, linguistic issues, and legal results.

Finally, present your recommended name, and demonstrate how it meets the objectives and criteria. You may wish to present some mocked-up packaging or product brochures that use the name prominently. Repeat the name in your presentation as many times as possible. Your goal here should be to help management feel as comfortable as you do, both with the fairness and professionalism of the naming process and with the name itself.

Once the name has been approved by management and before you share it much more extensively, you should ask your attorneys to file an "Intent-to-Use" application. This will reserve the name for your use.

Actual name roll-out should be coordinated with all other aspects of your product announcement and introduction plans. In conjunction with PR, promotion, and advertising activities, your new name should be prominently featured in all product introduction events.

The final step: Pat yourself and the naming team on the back for a job well done. You've done everything possible to give your product a name that will set it apart and help it reach its full potential in the market. ♣

SB Master is President of Master-McNeil, Inc., a firm located in Berkeley, California, that specializes in product and company naming and nomenclature systems.

Kinds of Names

There are two basic categories of names: names that are real words (Apple, Sun) and those that have no intrinsic meaning (Exxon, Abex). In between these extremes are a wide variety of hybrids with more or less intrinsic meanings.

Which category of name is right for your product depends on the qualities you need to convey, who your target market is, how crowded your product category is, and the size of your marketing budget. The following paragraphs describe some things to think about as you consider what kind of name will work best for your product:

Real-Word Names Many companies use real-word names for their products. There is a single-word form of this (Apple, Radius) and a combined form (AppleTalk, ColorSync, After Dark, HyperCard). These real-word names have the advantage of being immediately recognizable and understood, and go a long way in establishing what a new product has to offer. Well-chosen real-word names use simple vocabulary, so they usually retain their meaning worldwide, no matter what language they are built upon.

The problem with real-word solutions, however, is that the more descriptive, relevant ones are often already being used by someone else. The less descriptive, less relevant ones are less desirable, because you need to spend time and money to help explain, establish, and attach an image to them. However, real-word names are worth a lot of serious creative exploration. If you can find a relevant real-word solution that is unique and effective, it will be among the easiest of names to establish it in the marketplace.

Coined Names At the other end of the spectrum are made-up or "coined" names. As the number of new products increases, finding an available real-word name solution is becoming increasingly difficult. So, companies are increasingly turning to coined names to identify their products.

Coined names can be subdivided into arbitrary/obscure and meaningful categories. Some examples of the arbitrary might include Synovus Financial, Allegis Corp., Ceridian (the Control Data spin-off), and perhaps Pentium. These are all examples of coined words which on their own mean little, if anything, to anyone. However, they can work. One way is to spend a lot of money establishing them. One example might be Exxon.

Coined names that work best, however, are those built on large enough "chunks" of language that the resulting names have intrinsic meaning. The best of these combine intrinsically meaningful, relevant roots into whole new words. Some examples are Navistar, Unisys, and Televangelist.

Why are these kinds of solutions worth pursuing? Primarily because of legal realities. As we discussed in the main body of the article, it is becoming more difficult to get legal approval for new names, and software seems one of the most difficult categories of all. We believe the latter group of names, the intrinsically meaningful "coined" ones, will be the place where many companies increasingly find candidate names that pass legal scrutiny. ♣

GetNextEvent

The indicates the trade shows/events at which Apple Computer, Inc., is scheduled to exhibit as of press time. This list may be incomplete. If you have information about a show that you want listed here, contact Developer Technical Communications, 20525 Mariani Avenue, M/S 75-3B, Cupertino, CA 95014. For further information check the Events folder on AppleLink (path—3rd Party Connection:Events).

September 14 through 20
** Sonimag**
Barcelona, Spain
Contact: Yolanda De Juan
AppleLink: SPA.EVENTS

September 15 through 18
** Apple Expo Paris**
Paris, France
Contact: Catherine Massot
AppleLink: FRA.PROMO
33-1-6986-3620

September 20 through 24
** Data Kontor**
Stockholm, Sweden
Contact: Per Hedlund
AppleLink: HEDLUND1

September 26 through October 1
** Oracle**
Orange County
Contact: Russ Havard
AppleLink: HAVARD1
(408) 974-4371

October 17 through 20
** EDUCOM**
Cincinnati, OH
Contact: Eliza Lapé
AppleLink: ELIZA
(408) 974-1248

October 18 through 22
FY '94 Global Sales Conference
Las Vegas, NV
Private event by invitation
Contacts:
Marva Whelan (408) 974-8561
Gail Bridges (408) 974-3094
Al Hoodwin (408) 974-4419
Thad Carhart 33-1-4901-4720
AppleLink: WHELAN2, BRIDGES2 (Apple USA), HOODWIN (Apple Pacific), CARHART (Apple Europe)

October 27 through 29
** ITTE/NSBA**
Dallas, TX
Contact: Jeryl Gerhardt
AppleLink: JERYL
(408) 974-2368

October 28 through 30
** Seybold**
San Francisco, CA
Contact: Russ Havard
AppleLink: HAVARD1
(408) 974-4371

November 7 through 10
MacIS Conference "Back to the Future"
Orlando, FL
Contact: Jerry Star
AppleLink: JERRY.STARR
(408) 974-3836

November 12 through 16
** National Alliance of Realtors (NAR)**
Miami Beach, FL
Contact: Cheryl Bunch
AppleLink: BUNCH1
(408) 974-2853

November 14 through 17
** League for Innovation**
Nashville, TN/Apple in attendance
Contact: Jeryl Gerhardt
Apple Link: JERYL
(408) 974-2368

November 15 through 19
** Comdex**
Las Vegas, NV
Contact: Dave Billmaier
AppleLink: BILLMAIER1
(408) 974-6553

APDA Ordering Information

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