

Inside this issue

Welcome to the electronic version of *Apple Direct*. Elsewhere in the *Apple Direct* folder on this CD are all the articles contained in the paper version of the October issue of *Apple Direct*. The guide below will serve as a table of contents and direct you to the appropriate subject folder: "News," "Marketing/Business," "New Product Intros," or "Tog."

NOTEBOOKS, NEW 040 CPUS, and other new Apple products are highlighted in this month's main technology story.

(See the New Product Intros folder)

TO KEEP UP ON APPLE/IBM developments, you can now access an AppleLink folder with read-only updates and participate in an online AppleLink discussion.

(See the News folder)

APPLE JOINS THE SPA Software Protection Fund and takes other steps towards launching an antipiracy campaign.

(See the News folder)

WONDERING HOW TO NAVIGATE the new newspaper version of *Apple Direct*? We'll tell you what's changed, what hasn't, and how to find what you need.

(See the News folder)

QUICKTIME APPS are in demand at Apple. If you have an application that uses QuickTime, Apple wants to know about it. Send a Link to the

COOL.APP AppleLink address. Also, there's now an AppleLink discussion board for QuickTime developers.

(See the News folder)

NEW APDA PRODUCTS that have been released in recent weeks will now be listed in Apple Direct in the “Now Available from Apple” section. Look there for pointers to information about new system software releases as well.

(See the News Folder)

CAN'T FIND THAT PAST Tog column that you'd like to refer back to? You can now find archives of his “Human Interface” column on AppleLink under the “Collected Works of Tog.”

(See the News Folder)

LOOK ON THE CD for the latest releases of Macintosh system software—including international versions.

(See the News Folder)

WHAT'S NEW ON THE CD? Find out by reading the “Developer CD Highlights” section each month in *Apple Direct*. You'll find the debut of this column in this issue. This month's CD features new product data sheets, international versions of System 7.0, cool hacks, and more.

(See the News Folder)

THE MAJORITY OF MAC development teams have between three and 12 employees, according to an Apple survey of its U.S. commercial developers.

(See the News Folder)

MACINTOSH DEBUGGING is the subject of a new two-day course from Apple's Developer University.

(See the News Folder)

THE DEV TECH ANSWERS LIBRARY on AppleLink has the latest technical information. Check your CD first to get answers to your technical questions, then log on to Link to see if there's more recent information uploaded there.

(See the News Folder)

EVEN TOG RUNS INTO difficulties when designing new human-interface objects. This month he shares his personal experiences with the Big Three enemies of software design: rationalization, assumption, and denial.

(See the Tog folder)

QUARK INC. SHARES its views on addressing the high end of the market in this month's "Developer Outlook." By aiming high, Quark finds that it also pulls in less-sophisticated customers as well. The keys to success are modularity and keeping an ear open to the high-end customer's needs.

(See the Marketing/Business folder)

GETTING REVIEWS that show your product in the best light takes a lot of fact-finding and leg work. John Pearce, CEO of MediaMap, tells how to get the attention of review editors and influence them favorably.

(See the Marketing/Business folder)

THE PACIFIC MARKET Forum 1992 will take place in San Francisco just after Macworld Expo in January.

(See the Marketing/Business folder)

THIS MONTH'S EVENT calendar lists conferences, trade shows and other events through December, 1991.

(See the Marketing/Business folder)

MACS WITH FOUR MEGS are becoming the norm. In a recent Apple survey of Macintosh users, almost 40 percent of the respondents said they used systems with four or more megabytes of RAM, up from 22 percent two years ago.

(See the Marketing/Business folder)

IF YOU'VE ALWAYS wanted your product listed in the It Shipped section in *Apple Direct*, this month's column includes instructions on how to participate.

(See the Marketing/Business folder)

A Guide to the New Apple Direct

If you're a regular or even occasional *Apple Direct* reader, you've undoubtedly noticed that the format has undergone a pretty dramatic change. The redesign of the paper version is one of two major changes to the Developer Monthly Mailing to take effect this month.

The other change is a shift to a monthly Developer CD schedule (rather than quarterly, as it has been in the past). In order to simplify the mailing, it will now contain only two items—the CD and *Apple Direct*—with additional items such as videos and handbooks included in the mailing only occasionally.

If you'd like additional subscriptions to the monthly mailing for other people in your company, you can purchase subscriptions through APDA. See "Now Available from APDA," in the News folder, for information on contacting APDA.

As you browse through the paper issue, you'll see that *Apple Direct* is now divided into two sections, which contain almost all the previous columns the magazine format contained—and then some. The front section will cover news and technology stories, and the second section is devoted to business and marketing material. Here's the nickel tour to the new *Apple Direct* and how it relates to the old:

- *Technology/new-product feature stories.* You'll see that the first newspaper issue has several pages of new-product coverage, starting on page 1 and continuing inside. When we run technology or new-product features, they'll always begin on the front page of the newspaper.
- *News.* One of the main advantages of going to a newspaper format is that it allows us to expand our news coverage and also to shorten our lead time so that we can get information to you more quickly. News will start on page 1 and take up the first few inside pages of the front section. If there's a news item that relates specifically to business or marketing, though, it may appear in the Business and Marketing section.
- *Tog.* If you're a fan of Bruce Tognazzini's Human Interface column, you'll always be able to find it quickly by looking at the back page of the first section. In this issue, however, his column is inside the front section (see page 9) to allow for a special double-length column.
- *Latest Rev.* Latest Rev is one of the few sections from the previous incarnation of *Apple Direct* that won't appear in the new one. Instead, "Latest Rev"—which lists the latest version numbers of Apple system software and where to obtain it—will now live on the CD, in the Information Resources folder. Replacing it is a new section called "Now Available from Apple," which lists system software and APDA products that have been

released in the past month (see column at left). “Now Available from Apple” will always appear in *Apple Direct*’s front section, along with the news.

- *Developer CD Highlights*. This new column is your guide to what’s new on the current month’s CD. It will always appear on page 3.

- *Marketing features*. How-to marketing articles will be the mainstay of the new Business and Marketing section (see page 13). You’ll always find them starting on the front page of this section.

- *Developer Outlook*. This developer-written feature, which covers a different business issue each month, will also always start on the front page of the Business and Marketing section.

See page 13 for this month’s “Developer Outlook” about the high-end market, written by Peter Warren, of Quark, Inc.

- *Market Insight*. We’ve converted this column from an article-length feature to a chart-of-the-month format, making it more convenient for you to skim and absorb the information. “Market Insight” will appear in the Business and Marketing section (see page 14 in this issue).

- *Metric of the Month*. Besides data on the Macintosh market, we’ll also be bringing you a chart each month that gives you data on other development-related issues (average development time, size of typical Macintosh development teams, time to market, and the like).

See page 3 in this issue for this new feature, which will always appear in the front section.

- *GetNextEvent*. Our popular events calendar will continue each month, in the Business and Marketing section (see page 13).

- *It Shipped*. We will be expanding It Shipped in the coming months to include non-U.S. companies and other international data. It Shipped will alternate between the first and second sections in *Apple Direct*.

Remember too that beginning next month, *Apple Direct* will be archived on the CD, so you can toss the hard copy into the recycle bin or pass it on when you’re done with it and still have access to past *Apple Direct* articles.

We welcome your feedback on the new format, the new mailing, and the availability of the *Apple Direct*/Developer CD package through APDA. Send your comments, via AppleLink, to RALEIGH.L. ◇

Apple/IBM Update

On October 2, Apple and IBM announced that they had finalized the terms of the technology alliance first revealed in a letter of intent on July 3, 1991. The alliance applies to five specific development areas: interoperability, UNIX, RISC, multimedia, and object-based systems.

To help Apple developers stay up-to-date on information relating to the alliance, and to encourage developer discussion of the topic, we've established two areas on AppleLink:

- The Apple/IBM Alliance folder contains press releases, Q&As, and other materials in a read-only format. As more details become available, this is where you'll find them. The AppleLink path is Developer Support: Developer Services: Apple/IBM Alliance.
- The Apple/IBM Alliance Discussion provides a forum for developers to discuss the alliance with each other. The AppleLink path is: Developer Support: Developer Talk:Apple/IBM Alliance Discussion.

As the alliance continues to evolve we will keep you updated via *Apple Direct* and AppleLink. Please stay tuned... ◇

The October Revolution

by Gregg Williams, Apple Direct staff

Notebooks, 68040s, printers, etc.

October has been one tremendous month for new Apple products. First, there was the unveiling of the LaserWriter IIf, LaserWriter IIg, OneScanner, and Macintosh 21" Color Display at the Seybold Computer Publishing Expo in San Jose, California.

Then the introduction of AppleShare Server 3.0 and AppleTalk Remote Access at NetWorld, in Dallas, Texas.

And finally the worldwide introduction of three notebook computers (the PowerBook 100, 140, and 170), two 68040 computers (the Macintosh Quadra 700 and 900), and the 68030 Macintosh Classic II.

In one month, Apple has introduced more products in more categories than it did in the first few years of the Macintosh's nearly eight-year history. And each of these products either brings something entirely new to Apple's product line or delivers more useful computing or printing power for less money—and some do both.

The rest of this article describes no fewer than 12 products in 8 different categories. The fact sheets scattered throughout the article give overviews of each product (including the preliminary prices in effect when this article went to press).

Look to the sections below for selected information on each product, especially details you need to know as a developer.

Macintosh PowerBook 100, 140, and 170

To rephrase an old riddle: "What are the three most important things about a portable computer?" The answer is, of course, "Weight, weight, and weight." (*Computing power and size* come immediately thereafter.)

I don't know exactly where it happens, but when a computer shrinks from 15 pounds (the Macintosh Portable) to 5–6.8 pounds (the new PowerBooks), it makes the fundamental transition from *moderately useful* to *incredibly essential*.

Imagine a Macintosh Portable, shrunk to 5 pounds—that's the \$2,499 Macintosh PowerBook 100, which includes an external SuperDisk floppy-disk drive. (All the prices in this article are U.S. Suggested Retail Prices, as they stood as this article went to press.) Transmute a Macintosh IIcx to weigh in at 6.8 pounds, and you have the \$4,599 top-of-the-notebook-line PowerBook 170.

Take another notebook with about two-thirds of the power (and price) of the PowerBook 170, and you have the \$2,899 PowerBook 140. And straight out of the box, all three are powerful System 7 machines (they use System 7.0.1).

I can speak from my own experience: Once you've gotten your hands on one of these, you won't want to give it up—and we expect that millions of people will feel the same way.

Why “PowerBook”? Yes, PowerBook is an odd name, but in one word, it makes two important points. First, *Power* implies that these machines have the power and ease of use of a desktop Macintosh computer. (I certainly found this to be the case when I worked with both the PowerBook 100 and the 140. I found the quality of their backlit video displays and keyboards so good that I used both of them for hours without any sense of strain or compromise.)

Second, *Book* says that you get that uncompromisingly Macintosh experience in the size and weight of a notebook computer. I really believe that the combination of these two factors will make the Macintosh PowerBooks irresistible.

(On the other hand, you may ask “Why the numbers 100, 140, and 170?” Well, there are some things mere mortals were not meant to know....)

The PowerBook 100 is based on the Motorola 68000 processor, whereas the PowerBook 140 and 170 use the 68030. Table 1 summarizes the features of all three. The sections below deal with the PowerBook 100 first and then the PowerBook 140 and 170.

In the “PowerBook 100 Features” section, I'll also be pointing out features that belong to all three PowerBooks.

The Only Notebook Some May Ever Need. Let's face it—most people in this world don't need a Macintosh IIcx (although it's certainly OK to *want* one). They use a Macintosh for word processing, spreadsheets, and maybe a database or a scheduler/calendar. The PowerBook 100, which has twice the power of our entry-level Macintosh Classic, is certainly up to such tasks.

Add the fact that it's light enough to take it wherever you go (a very reasonable 5 pounds), and you've got a Macintosh that is ideal for a very large section of current Macintosh owners and an even larger number of people who have never—until now—owned a computer of any kind.

In the Macintosh PowerBooks, ease-of-use is built-in. You need only look at all three PowerBooks (see figure 1) to see that they don't look like any other notebook computer around.

We didn't put the trackball below the keyboard and the palmrests on either side of it just to look different. Their presence and position are the result of much human-factors design and testing. Two "lifters" on the rear two corners of all the PowerBooks allow you to tilt the keyboard toward you slightly.

I used the PowerBook 100 for several days and found the keyboard/palmrest/trackball configuration very easy to use. In fact, the PowerBooks are easier than other notebooks to use in your lap because the keyboard isn't jammed up against your stomach. (This, I assume, would also be true for airplane tray tables.)

The video displays on all three PowerBooks (discussed below in the PowerBook 140 and 170 section) are backlit and very easy to read.

Another way that ease of use is built in to the PowerBook series is that these three computers, like all other Macintosh models, make it easy for users to communicate and exchange data with each other.

PowerBook 100

Microprocessor

- 68HC000, running at 16 MH
(2 times faster than Macintosh Classic)

Memory

- 2 MB of pseudostatic RAM (expandable to a maximum of 8 MB via single memory expansion slot)
- 256K of ROM

Disk Drives

- Internal 20-MB SCSI hard disk
- External 1.4-MB low-power Apple SuperDrive

Display

- 9 in. (229 mm) diagonal backlit Supertwist liquid crystal display
- 640 by 400 pixels

Battery

- Sealed lead-acid; provides 2 to 4 hours of usage before recharge needed

Interfaces

- One ADB port
- One serial (RS-422) port
- One HDI-30 SCSI port
- One monaural sound output port
- One HDI-20 floppy disk drive port

Other

- Keyboard with 25 mm-diameter trackball
- Optional internal 2400-baud modem with fax send at 9600 baud (\$349)
- Line voltage: 110 to 220 volts at 50 to 60 Hz
- Weight: 5.1 lb. (2.3 kg)
- Clock/calendar chip and battery
- Battery and AC adapter
- System Software 7.0.1
- Documentation and training software
- Contains no processor-direct slot (PDS) or cache-card connector

U.S. Suggested Retail Price

- \$2,499 (includes 2 MB RAM, 20-MB hard disk, external floppy disk drive)
- \$2,299 (without floppy drive)

PowerBook 140**Microprocessor**

- 68030, running at 16 MHz
(3.5 times faster than Macintosh Classic)

Memory

- 2 MB of pseudostatic RAM (expandable to up to 8 MB via single memory expansion slot)
- 1 MB of ROM (32-bit clean)

Disk drives

- Internal 20- or 40-MB SCSI hard disk
- Built-in 1.4-MB SuperDrive

Display

- 10 in. (254 mm) diagonal backlit Supertwist liquid crystal display
- 640 by 400 pixels

Battery

- NiCad; provides 2 to 3 hours of usage before recharge needed

Interfaces

- One ADB port
- Two serial (RS-422) ports
- One HDI-30 SCSI port
- Monaural sound-in and sound-out ports (includes microphone for sound input)

Other

- Keyboard with 30-mm diameter trackball
- Optional internal 2400-baud modem with fax send at 9600 baud (\$349)
- Line voltage: 110 to 220 volts at 50 to 60 Hz
- Weight: 6.8 lb. (3.03 kg)
- Clock/calendar chip and battery
- Battery and AC adapter
- System Software 7.0.1
- Documentation and training software
- Contains no processor-direct slot (PDS) or cache-card connector

U. S. Suggested Retail Price

- With 2 MB memory and 20-MB hard disk—\$2,899

- With 2 MB memory and 40-MB hard disk—\$3,199
- With 4 MB memory and 40-MB hard disk—\$3,499

PowerBook 170

Microprocessor

- 68030, running at 25 MHz
(5 times faster than Classic)
- 68882 math coprocessor,
running at 25 MHz

Memory

- Same as PowerBook 140,
except 2 MB of pseudostatic RAM (main memory) on the logic board, 2 MB in
expansion
card—for a total of 4 MB

Disk drives

- Internal 40-MB SCSI hard disk
- Built-in 1.4-MB SuperDrive

Display

- 10 in. (254 mm) diagonal
backlit active-matrix liquid
crystal display
- 640 by 400 pixels

Battery

- Same as PowerBook 140

Interfaces

- Same as PowerBook 140

Other

- Same as PowerBook 140, except modem is included

U.S. Suggested Retail Price

- With 4 MB memory, 2400-baud internal modem, and 40-MB hard disk—
\$4,599

Table 1: Fact sheet for the Macintosh PowerBook 100, 140, and 170

When you are close to the network you use, all you do is plug a network node into the back of your PowerBook and you can use features such as Macintosh file sharing and alias icons to obtain easy access to files anywhere on the network. When you are physically distant from your network, you can still get full access to your network through a truly remarkable product, bundled with each PowerBook, called Apple-Talk Remote Access. For details, see the section on it near the end of this article.

PowerBook 100 Features. To elaborate on a statement I made earlier, the PowerBook 100 is almost completely equivalent to the Macintosh Portable in both software and hardware. So there's not much that you, as a developer, need to know about the PowerBook 100 except that if your program runs correctly on the Macintosh Portable, it will run fine on the PowerBook 100.

Still, there are several interesting points about the PowerBook 100 that deserve mention:

- *Memory:* The PowerBook 100, 140, and 170 each have 2 megabytes of pseudostatic memory on their motherboard and an internal memory-expansion slot; the PowerBook 170 has that slot filled with a 2-megabyte memory card, for a total of 4 megabytes.

With 2- and 4-megabyte memory cards available from Apple and other sizes from third-party vendors (theoretically, a memory card can contain 1 to 6 megabytes in 1-megabyte increments), you can increase these three PowerBooks to a maximum of 8 megabytes of memory apiece.

- *Detachable 1.4-megabyte SuperDrive:* The PowerBook 100 itself includes a 20-megabyte internal hard drive and a 1.3-pound external low-power SuperDrive. (You can also order the PowerBook 100 without the external drive, but the standard configuration—the one that Apple recommends—is the PowerBook 100 with its external SuperDrive.)

Granted, I was originally a bit squeamish about the idea of a notebook computer without a floppy drive, but I now think of it as a way to make the notebook a pound lighter when you don't need it. (After all, you do have an internal hard disk on which to store programs and documents. It's not as if you had to store documents only in memory and trust the battery not to fail—which is where I *do* draw the line.)

The PowerBook 100 is the only PowerBook that has a port on its rear panel for an external floppy drive.

- *SCSI disk mode*: OK, if you want to shave \$200 off the price of the PowerBook 100, you can buy it without the external floppy drive. How, then, do you get files into and out of it (aside from using a modem or connecting the notebook to an AppleTalk network)? The PowerBook 100 is the only one of the three models that has a *SCSI disk mode*.

By using a special cable, you can connect a PowerBook 100 to the end of your deskbound Macintosh's chain of SCSI peripherals. (This is a bit of a bother, since both computers must be turned off before being connected.) When you turn both computers back on, the PowerBook 100's hard disk appears on the deskbound Macintosh's desktop as another SCSI hard disk.

This gives you an easy and efficient way to move files to and from the PowerBook 100, back up its internal hard disk, and install new software on it (from floppies inserted in the deskbound Mac).

- *The keyboard*: The keyboard on all three PowerBooks conforms to the ANSI standard (in the domestic version) or the ISO standard (in the European version). If you can hardly live without an Apple Extended Keyboard (as is the case with me), you will miss the numeric keypad and function keys when you use a PowerBook keyboard. If you really want to, though, you can connect an external ADB keyboard and/or mouse via each Power-Book's ADB port, but they will consume more power than their built-in equivalents and decrease the PowerBook's battery life.

One interesting note: The Caps Lock key doesn't mechanically lock down, so PowerBook system software keeps track of its state and displays an arrow icon near the right end of the menu bar when the Caps Lock key is "down."

- *Rear-panel connectors*. Because of space limitations, Apple had to use a small, square connector called an HDI-30 on all three PowerBooks for its SCSI port. (The PowerBook 100 uses a similar connector, called an HDI-20, for its external floppy drive.)

One of the few features of the Macintosh Portable that the PowerBook 100 lacks is a second external serial port: The PowerBook 100 has only one, which is used for connection to a LocalTalk network or a printer.

- *Internal modem:* The PowerBook 100 and 140 have room for an optional internal modem, and the PowerBook 170 comes with a 2,400-bps modem built in and software that lets you send faxes at 9,600 bps. You can connect the modem to a telephone line through an RJ-11 connector on the back of the PowerBook.

Although the PowerBook 100 has only one serial port on its back panel, you can still be simultaneously connected to both a modem (through the internal modem and the RJ-11 plug) and a printer/network (through the single serial port).

- *Sound out:* All three PowerBooks have a built-in speaker and an external stereo headphone jack that plays a monaural signal to both ears.

PowerBook 140 and 170 Common Features. The PowerBook 140 and 170 differ from the PowerBook 100 in that these two use a Motorola 68030 processor (instead of a 68000), they have a built-in SuperDrive, and they weigh 6.8 pounds (instead of 5). Here are the other major features they share:

- *ROM:* Both the PowerBook 140 and the 170 contain a 1-megabyte ROM with an access time of 150 nanoseconds. The ROM is also 32-bit clean, which means that if your program is 32-bit clean, your customers can run both it and their PowerBook 140 or 170 in 32-bit mode.

- *Shutdown mode:* In addition to the Sleep menu item (which sustains most of the internal electronics at a low current drain), the PowerBook 140 and 170 implement a true shutdown action. (The shutdown feature in the Macintosh Portable shuts down very little in the computer and consumes roughly the same amount of power as sleep mode does.)

In the PowerBook 140 and 170, Shutdown mode turns off most of the computer—the main memory; the custom integrated circuits; and the chips that support the keyboard, Power Manager, disk drives, and the SCSI and serial ports—leaving current to only a few subsystems such as the real-time clock, parameter RAM, and a few essential support circuits. Shutdown mode uses about 4 percent of the current needed by the sleep mode and, if used, gives a PowerBook 140 or 170 a much longer battery storage life between recharges.

- *Power cycling*: When the Macintosh Portable is not doing any active work, it increases the number of main-memory wait states from 1 to 64. In both the Portable and the PowerBook 100, this is called the Idle mode; it saves about 5–10 percent of the power the computer would otherwise be using.

The PowerBook 140 and 170 do something called *power cycling*, which saves about 15 percent of their power consumption. Power cycling begins when system software has detected no input activity for approximately two seconds. It then saves the registers of the 68030 processor (and, in the PowerBook 170, the 68882 math coprocessor) and turns off power to the processor (and, in the PowerBook 170, its math coprocessor).

Power cycling returns power to the processor(s) sometime before 16 milliseconds has expired—in other words, power is off for a set fraction of every 16-ms cycle. The power-cycling software then restores the registers and checks for activity. If the software finds activity, it services it and waits for the two-second time delay that initiates power cycling.

If it finds no activity, it repeats the power-saving cycle immediately, leaving power off for a larger fraction of the 16-ms cycle. In all cases, the PowerBook 140 and 170 maintain power to the video display, scan the keyboard, and keep the cursor blinking.

- *Sound input*: Like the Macintosh LC and Macintosh IIsx before them, the PowerBook 140 and 170 have sound input built in to them. (The Macintosh Classic II, described below, also has it. The PowerBook 100 doesn't, largely because the Macintosh Portable, after which it was patterned, doesn't.)

Apple has been putting sound input into its computers as one step toward making the Macintosh a platform that gives its users multiple ways to express themselves—color graphics, text, sound, and (soon, with QuickTime) animation and video.

- *Battery*: Apple engineers were able to reduce the weight of the PowerBook 140 and 170 batteries and increase their capacity slightly by moving from a lead-acid battery (used by the Macintosh Portable and PowerBook 100) to a nickel-cadmium (NiCad) battery.

Historically, NiCads are known to lose power very quickly just before total discharge, thus presenting the possibility that the battery might fail before it could warn you and let you save your data. NiCad batteries are also said to have a “memory” problem in which their electrical capacity decreases if they are not fully discharged before being recharged. However, Apple engineers did

extensive testing of the NiCad batteries that are used in the PowerBook 140 and 170 and found the following:

First, these two PowerBooks—like the PowerBook 100—can detect battery exhaustion in plenty of time to warn users, and second, the capacity loss of these NiCad batteries is not that big a problem—on average, less than 10 percent of total capacity (some NiCads used by other vendors still have a problem with capacity loss).

PowerBook 140 and 170 Differences. Several distinctions between the two make the PowerBook 170 the high-end model. Here are the main ones:

- *Processor speed:* The PowerBook 140's 68030 processor runs at 16 megahertz, whereas that of the PowerBook 170 runs at 25 megahertz. This makes the PowerBook 170 run about 50 percent faster than the PowerBook 140. In fact, the PowerBook 170 is roughly equivalent to a monochrome Macintosh IIfx—but one that weighs 6.8 pounds. (Wow!)



Figure 1: The Macintosh PowerBook 100, 140, and 170 computers.

- *Math coprocessor:* The PowerBook 170 contains a Motorola 68882 math coprocessor (also called a floating-point unit, or FPU), running at 25 megahertz. The PowerBook 140, on the other hand, does not include one, nor was it designed to have one added. The 68882 contributes to the PowerBook 170's speed, especially in applications that do a lot of number crunching, such as spreadsheet programs, page-layout programs, and CAD packages.

- *Video display:* Both the PowerBook 100 and the 140 use a backlit Supertwist display (also called *Film Supertwist Nematic*, or FSTN). On these two PowerBooks, you can control the brightness of the backlighting and the overall contrast by using two controls located below the display.

The PowerBook 170 uses a backlit active-matrix display (also called *Thin-Film Transistor*, or TFT). The PowerBook 170 uses exactly the same display as the current (backlit-screen) Macintosh Portable—and the result is beautiful; it needs only one control, for brightness. The Supertwist displays on the PowerBook 100 and 140 are also very readable.

- *A 40-megabyte internal hard drive:* The PowerBook 140 comes with a 20- or 40-megabyte internal hard disk and the PowerBook 170 comes with a 40-megabyte internal hard disk. Apple is working with third-party vendors to create other internal hard drives for use with the PowerBook 100, 140, and 170. (Future configurations of these three PowerBooks will contain 40- and 80-megabyte internal hard drives.)

- *Power Saver mode:* On the PowerBook 170 only, you can open the Battery program and turn the Power Saver mode on. When you reboot, the PowerBook 170 runs at 16 megahertz. This cuts down on power consumption, which means that if you're running on the PowerBook 170's battery, it should last about 30 minutes longer than it would have if it had been running at its normal 25 megahertz.

Developer Opportunities. The PowerBook 100, 140, and 170 offer two kinds of developer opportunities: first, new hardware and accessory products and second, new kinds of software and enhancements to existing software.

You may want to investigate creating new internal modems and hard drives for the PowerBook line. Be aware that all internal enhancements have strict, we-really-mean-it-this-time size and power-consumption constraints. You can consult the PowerBook's *Developer Notes* for details.

(A document tentatively entitled the *Macintosh Classic II, Macintosh PowerBook Family, and Macintosh Quadra Family Developer Notes* should be available from APDA sometime in November 1991. To contact APDA, see the “Now Available from Apple” column on page 2.)

There will be a market for 1-to-6-megabyte memory-expansion cards. When designed correctly, the same card will work in the PowerBook 100 (which has a 16-bit data path), and in the Power-Book 140 and 170 and future PowerBook models (all of which have 32-bit data paths). Check the PowerBook’s *Developer Notes* for details.

PowerBooks will be used by many people who have never owned a computer. They will also be used (by new or existing computer owners) in new ways—in particular, people will be carrying their PowerBooks with them all the time. With these new situations come possibilities for new kinds of software.

Obvious possibilities include personal-information-manager programs, programs for salespeople on the road, and data-transfer programs (to help move data between PowerBooks and deskbound Macintosh computers). And, of course, we rely on you to think of creative new software that will make PowerBooks even more indispensable.

Etiquette for Power-Conserving Computers. Your program shares responsibility with Power-Book software and hardware for saving as much power as possible. Code that is correct for previous Macintosh computers may waste power and potentially even damage the PowerBook 100.

Note that power cycling automatically turns off certain parts of the PowerBook and that shutdown (by the user or the PowerBook) turns off even more. Here are some things of which you should be aware:

- Do not automatically open AppleTalk or the serial driver unless you are going to use them. While they remain on, a PowerBook loses more power than it would otherwise.

More importantly, if the PowerBook is left unattended and the battery nears exhaustion, the PowerBook (thinking that a network connection is active) will not power itself off—which will exhaust the battery. (In the PowerBook 100, if the lead-acid battery power gets too low, it becomes *sulfated* and cannot be recharged.) Also be aware that animated cursors prevent power cycling.

- Hard-coded timing loops are a bad idea in general; here they’re even worse. Use system timing constants such as TimeDBRA and TimeSCCDB instead.

- To system software, power cycling looks like a background task waiting to be serviced by `WaitNextEvent`. So when your program doesn't need system resources, it should call `WaitNextEvent` (which it should be doing anyway). This lets power cycling occur and also gives other programs a chance to run.

- Wherever possible, get the data you need from the hard disk (or write it to disk) all at once. When the hard disk goes to sleep, it takes a lot of power to get it spinning at full speed again.

Compatibility Issues. The same prohibitions that have been true on other Macintosh models are still true, but they bear repeating:

- Don't interact directly with the hardware; use the routines provided by the system software. In several cases, the chip your program is looking for no longer exists and its functions are contained in another chip (or chips).

- Don't assume that just because the PowerBook 140 is a 68030-based machine, a 68882 math coprocessor is available. The 68882 is present on the PowerBook 170 only. The PowerBook 140 and 170 return the same Gestalt machine-selector code, so you must check for the presence of the 68882 to determine which PowerBook is present.

(In general—and this is *very important*—you should not use Gestalt to check for a particular Macintosh model name and infer features from that; instead, you should check for the presence or absence of individual features. As Apple continues to add new products, testing for features makes more and more sense.)

- Like other Macintosh models, the PowerBook 140 and 170 both record and play back monaural sound. If you use the Sound Manager routines, everything will be fine. But if you access the sound hardware directly (which you shouldn't do), make sure you send your sound data to the *left* channel—you will lose any data you send to the right channel.

Because each of the three PowerBook models uses System 7.0.1, they do not support the (old) Sound Driver software or the (older still) wave-table sound-output mode. You should use the Sound Manager routines instead.

- Because of their heritage from previous Macintosh models, the PowerBook 140 and 170 ROMs contain the Slot Manager routines, even though they have no NuBus slots. So don't assume that just because your program detects the Slot Manager, it is running on a Macintosh that has slots.

A Macintosh to Take Along with You. When Jean-Louis Gassée was at Apple, he was reported to have said that all Apple wants is to sell every person on the planet three computers: one for work, one for home, and one to take with you. Although the Macintosh Portable was the first Macintosh “to take with you,” the Macintosh PowerBooks will give many, many people an excellent first experience with portable Macintosh computing.

And no matter how light and how powerful future Macintosh computers may be, many of us will look back to October 1991 with great fondness and say, “It started *here*.”

MACINTOSH QUADRA 700 AND 900

Things don’t stay the same for very long in the computer industry, so it’s time for the Macintosh IIx to step aside as Apple’s fastest computer and make way for the Macintosh Quadra 700 and 900, both of which can run software up to twice as fast as the Macintosh IIx.

As detailed below, the Macintosh Quadra 900 is very expandable, which means that you can configure it to be as powerful as you want it to be.

From the features listed in Table 2, you can see that the Macintosh Quadra 700 and 900 are very similar to each other in computation power. Apple believes that the Macintosh Quadra 700 will be the mainstream model that will provide an affordable entry into the 68040 market for many people. The Macintosh Quadra 900, on the other hand, has a larger capacity (in summary, more memory, NuBus slots, power per NuBus slot, and internal storage options).

This makes the Macintosh Quadra 900 the better choice for resource-intensive programs such as 24-bit paint programs, CAD packages, and ray-tracing graphics programs as well as for mainstream users who want the best performance possible.

Macintosh Quadra 700

Microprocessor

- MC68040, running at 25 MHz
- On-chip instruction and data caches (4K bytes each), PMMU, and FPU

Memory

- 4 MB RAM on logic board, 4 memory expansion slots for SIMMs
- Holds up to 20 MB using 4-MB SIMMs

Expansion hardware

- Two NuBus '90 slots
- One processor-direct slot (PDS)

Disk drives

- Built-in 1.4 MB Apple Super-Drive
- Optional internal 80-, 160-, or 400-MB SCSI hard disk

Video

- 512K of video memory, expandable to 2MB
- Internal video supports all Apple monitors

Interfaces

- Two ADB ports
- Two serial (RS-232/RS-422) ports
- SCSI interface, for external peripherals
- Video port
- Monaural sound-in and stereo sound-out ports
- Ethernet connector
- Built-in LocalTalk through serial port

Other

- Microphone, ADB mouse
- Keyboard not included
- Line voltage: 100-240 volts AC, at 50 to 60 Hz; 90-watt power supply
- Weight: 13.6 lb. (6.2 kg) (floppy-only model)

U.S. Suggested Retail Price

- Quadra 700, 4 MB RAM, no hard disk—\$5,699
- Quadra 700, 4 MB RAM, 80 MB hard disk—\$6,399
- Quadra 700, 4 MB RAM, 160 MB hard disk—\$6,999
- Quadra 700, 4 MB RAM, 400 MB hard disk—\$7,699 (available Jan. '92)
- Macintosh IIcx or IIci board upgrade—\$3,499 (available in Jan. '92)

Macintosh Quadra 900

Microprocessor

- Same as Quadra 700

Memory

- 4 MB RAM (4 1-MBSIMMs in 4 SIMMslots), 12 additional SIMM slots
- Holds up to 64 MB using 4-MB SIMMS in all 16SIMMslots

Expansion hardware

- Five NuBus '90 slots
- One processor-direct slot (PDS)

Disk drives

- Same as Quadra 700 except: no 80 MB internal hard disk offered; Quadra 900 supports 3 additional internal-storage devices; one can have removable media

Video

- 1 MB of video memory, ex-pandable to 2 MB
- Internal video supports all Apple monitors.

Interfaces

- Same as Quadra 700, except only one ADB port

Other

- Microphone, ADB mouse
- Keyboard not included
- Line voltage: 100-240 volts AC, at 50 to 60 Hz; 303-watt power supply
- Weight: 36.75 lb. (16.7 kg)(floppy-only model)

U.S. Suggested Retail Price

- Quadra 900, 4 MB RAM, no hard disk—\$7,199
- Quadra 900, 4 MB RAM, 160 MB hard disk—\$8,499
- Quadra 900, 4 MB RAM, 400 MB hard disk—\$9,199 (available Jan. '92)

Table 2. The Macintosh Quadra 700 and 900 Systems

Copyback Caching. Before we get into the main features of the Macintosh Quadra 700 and 900, you need to understand one very important feature of the 68040 processor—*copyback caching*—because the Macintosh Quadra 700 and 900 hardware architecture uses it exclusively and your programs must know how to deal with it.

Caching is a useful hardware technique for improving the overall speed of a processor. The processor stores recently accessed data in a special bank of memory called a *cache*. Later, if the processor needs the same data and can find it in the cache, it does so; because the cache is on the processor chip, retrieving data from the cache is faster than doing so from main memory.

In *write-through caching* (which is what most people are familiar with), data is simultaneously written to the cache and to its place within memory, thus ensuring that memory always contains the correct values.

In copyback caching, however, the data is initially written only to the cache; this is faster, but main memory is not always correct.

Eventually, the “fresh” data in the cache will update the “stale” data in main memory; this occurs when the processor or an executed instruction explicitly tells the cache to “flush” its contents back to main memory or when the cache automatically flushes its contents to make way for new data being written to the cache.

The use of a copyback cache can improve the 68040’s performance by up to 50 percent (over performance with a write-through cache). This is because normal code often writes new values to the same memory address in quick succession.

With a write-through cache, each write operation writes to both the cache and main memory. With a copyback cache, however, the processor writes to the cache multiple times (which it does quickly) but writes to main memory (an operation that slows the processor down) only once.

When a 68040 computer uses copyback caching—as the Macintosh Quadra 700 and 900 always do—the contents of memory are not always correct. This can cause subtle errors to occur if you don’t take copyback caching into account. See the “Compat-ibility and Programming Issues” section below for more details.

Macintosh Quadra 700 and 900 Common Features. Here are the most important features the Macintosh Quadra 700 and 900 share:

- *Built-in video:* Both Macintosh Quadra systems have built-in video that supports all Apple monitors, including the new Macintosh 21" Color Display. The Quadra 700 comes with a standard 512 kilobytes of video memory, which means it supports 8-bit (256 colors or shades of gray) on a 13" monitor and 4-bit (16 colors or shades of gray) on a two-page display. The Quadra 900 comes with a standard one-megabyte of video memory, which supports 8-bit color on all monitors and millions of colors on a 12" display. With the VRAM Expansion Kit, you can fully configure both systems to display even more levels.

Video memory is separate from main memory, and it is connected (through a custom chip) directly to the 68040 bus; both of these factors increase the speed of the video. The Macintosh Quadra video can also drive VGA, NTSC, and PAL monitors.

- *Built-in Ethernet support:* The Macintosh Quadra 700 and 900 both have Ethernet; the connector on the back panel supports thick, thin, and twisted-pair Ethernet cables (when used with the appropriate adapter).

- *Improvements to NuBus:* For the Macintosh Quadra 700 and 900, Apple designed the NuBus controller to be faster than before. Both computers implement the NuBus '90 interim standard, which can do NuBus '90 double-rate block transfers at twice the speed of the original NuBus.

(However, NuBus '90 defines four pins—/CM0, /CM1, /CM2, and /CBUSY—that are related to bus snooping [see the sidebar "The 68040: Motorola's Latest" for details]. Since the Macintosh Quadra 700 and 900 do not allow bus snooping, these lines are not implemented.)

The Macintosh Quadra 900 makes several more NuBus-related improvements; see the "Macintosh Quadra 700 and 900 Differences" section, below, for details.

- *Processor-direct slot:* Both computers have a processor-direct slot (PDS) that connects directly to the 68040 processor through the system bus. This slot can be used for add-on high-performance hardware cards that require direct access to the 68040.

Because PDS cards are tied to 68040 signals, PDS cards for these computers do not work in the PDS slots of other Macintosh models and vice versa.

Physically, the PDS slot is in line with the NuBus slot \$E and blocks access to it; you cannot use both at the same time. The PDS slot also maps directly into the memory space reserved for NuBus slot \$E.

- *Ω SANE*: The Macintosh Quadra 700 and 900 support Ω SANE, an enhancement of SANE (the Standard Apple Numerics Environment) that runs 2.5 times faster than standard SANE. When Ω SANE starts executing SANE code, it looks to see if the instructions are in memory (as opposed to ROM). If they are, it looks for certain sequences and replaces them with much faster code.

- *RAM-disk support*: Each Macintosh Quadra provides ROM support for activating a user-configured RAM disk that can survive a warm start and can, because of this, be used as a faster system-startup disk.

To prevent other programs from trashing the RAM disk's memory, the RAM disks' driver write-protects the memory pages that comprise the RAM disk and unprotects them only when it needs to make a (valid) change to the RAM disk. Because of this protection, you can reboot from the RAM disk even after a system crash or reset.

Macintosh Quadra 700 and 900 Differences. Although the Macintosh Quadra 700 and 900 are virtually identical in terms of computational ability, the Macintosh Quadra 900 adds enhancements that make it the better of the two for several heavy-duty situations. Here are the major differences:

- *Expandability*: The points below tell the details, but one of the Macintosh Quadra 900's biggest advantages is that you can expand it far beyond anything available on any other personal computer. A 32-bit-clean Macintosh application can access a 64-megabyte program space—more if you use Apple's Virtual-Memory feature. There's also room for three (or four, if you remove the internal SuperDrive) internal storage devices—like CD-ROM drives and very-high-capacity SCSI hard disks whose combined storage can easily exceed one gigabyte.

- *Memory*: The Macintosh Quadra 900 can currently contain as much as 64 megabytes of memory on the motherboard; like the Macintosh Quadra 700, the Macintosh Quadra 900 uses 4-megabyte SIMMs to expand past its 4-megabyte configuration. The Macintosh Quadra 700 is limited to (merely!) 20 megabytes.

- *NuBus-slot enhancements*: Besides the fact that the Macintosh Quadra 900 has five NuBus '90 slots (compared with the Macintosh Quadra 700's two slots), there are two other ways in which the Macintosh Quadra 900 NuBus slots are better for some uses. NuBus-card designers have often complained that previous Macintosh models supplied too little power for their NuBus slots. The Macintosh Quadra 900 fixes that with a 303-watt power supply (the Macintosh Quadra 700's is 90 watts). This allows the Macintosh Quadra 900 to power two slots at 25 watts each and three slots at 15 watts each (the same as other Macintosh models).

Another complaint about NuBus cards has been that the size Apple adopted was often too small for the products some developers had in mind. Because of this, Apple engineers designed the Quadra 900 to house oversized NuBus cards that are two inches taller than standard Apple NuBus cards. In addition, the Macintosh Quadra 900 NuBus slots have a STDYPWR (standby power) pin you may find useful; it supplies a low-power source of 5 volts that you can use to keep your card "alive" even when the Macintosh Quadra 900 has been turned off. So, all you ambitious NuBus designers, *dig in!*

(I have to mention two caveats: Apple says that it won't commit to using this oversized NuBus format in future Macintosh models. So if you use the oversized format, be aware that your products may be usable only on Macintosh Quadra 900 computers. Also, cards that use extra power may not work on other Mac models.)

- *CD-ROM playthrough*: On the motherboard, users can connect the audio output of a built-in CD-ROM drive to the Macintosh Quadra 900 sound subsystem, where it can play along with whatever internal sounds the Macintosh Quadra 900 makes. This also makes it easy for the sound-input subsystem to record sound from a CD-ROM.

Compatibility and Pro-gramming Issues. The Macintosh Quadra 700 and 900 are compatible with many applications. As always, though, Apple faces the challenge of pushing the technology of the Macintosh platform forward and retaining as much backward compatibility as possible. Here are the points, many of them already familiar to you, to keep in mind:

- Don't try to interact directly with the hardware; both models have several custom chips that replace the chips you think are there.

- Use the Gestalt routine to determine what features are present in the computer on which your program is running. (Don't determine the model on which your program is running and infer features from that.) This includes such things as the processor, the memory-management unit, and the math coprocessor. (Remember that the MMU and math coprocessor of the 68040 differ significantly from their earlier counterparts.)

In addition, the 68040 processor and other features of the Macintosh Quadra 700 and 900 present potential problems that your programs must address. These include:

- *Copyback caching*: Be sure you understand how copyback caching works before you have your program write data values that will later affect program execution. This includes writing data that will be executed as instructions (this includes but is not limited to self-modifying code and subroutine patches).

There are three keys to understanding copyback caching on the Macintosh Quadra 700 and 900. First, the 68040 processor has separate caches for data and instructions (4K bytes each) and they operate independently.

Second, the data cache caches read as well as write operations, whereas the instruction cache caches read operations only (you never write to memory through the instruction cache, only through the data cache).

Third, any data that goes into the data cache does not immediately appear in main memory. These three points may lead to totally unexpected program behavior. For more details, see Macintosh Technical Note #261, "Cache as Cache Can."

Another feature of copyback caching is that you are able to flush portions of the data cache and invalidate portions of the instruction cache. This allows you to do only necessary flushing and therefore improve your program's speed. To use this ability correctly, you need to understand the consequences of copyback caching.

The 68040 is Motorola's latest (and fastest) addition to its line of 680x0 line of processors. Like its predecessors, it is backward compatible with earlier models—meaning that it correctly executes existing object code meant for its predecessors. However, innovation always causes something to get done differently, and you should be aware of the differences.

Making Things Faster. To start off, the 68040 contains both a *memory-management unit* (or MMU) and a math coprocessor (also called a *floating-point unit*, or FPU). This contrasts with the 68030, which has an on-chip MMU but needs another chip, the 68882, to do mathematical computations in hardware. Given the right sequence of instructions, the 68040 can execute both integer and floating-point instructions in parallel. However, the 68040 math coprocessor does not execute all the instructions that the 68882 does (see main text for details). To maintain backward compatibility, code somewhere in the system must emulate these instructions. The Quadra's system software does this, and it is of necessity much slower than code executed directly on the 68882.

Motorola examined how people used the 68030 and optimized the things that 68030 did most frequently. For example, the integer unit (the subsystem of the 680x0 that performs integer arithmetic) now has a 6-step "pipeline." This means that the 68040 can simultaneously be working on up to six integer instructions, each in a different phase of its execution.

Motorola made several other changes that speed up the execution of instructions. The 68040's on-chip FPU has been optimized to run the most commonly used instructions faster. Also, it turns out that programs execute conditional branches (instead of "falling through" to the next instruction) about 75 percent of the time. Because of this, Motorola o p t i m i z e d t h e 68040 so that "branches" execute quicker than "fall-throughs." And the 68040 saves even more time by calculating both the "branch" and "fall-through" addresses of a conditional-branch instruction before the processor hardware decides which way to branch.

Caching Issues. The 68040 contains separate 4-kilobyte caches for both instructions and data. (The 256-byte caches on the 68030 proved to be too small to be overly useful.) Each cache and its MMU are on separate internal data buses, allowing the 68040 to use both of them simultaneously and independently. Each cache has its own *address translation cache*, which minimizes the time spent in translating logical to physical addresses.

Until now, 680x0 processors have used write-through caching to speed up multiple accesses to the same memory location. The 68040 adds copyback caching, which is approximately 50 percent faster than write-through caching but presents some problems, especially with existing code. The 68040 allows the designer to mix both write-through and copyback caching, but the Quadra 700 and 900 use copyback caching exclusively. See the "Copyback Caching" section in the main text for details of what this means for the Quadra 700 and 900.

No Snooping Allowed. The 68040 processor also allows *bus snooping*, which can be used when an alternate bus master reads from a copyback-cached page of memory. When bus snooping is allowed, the 68040 watches the alternate-master read operations from memory; if the alternate-master attempts to read a memory location that contains “stale” data, the 68040 intervenes and hands back the correct value from the cache. (The 68040 does something similar for write operations.)

However, the Quadra 700 and 900 do not use this feature of the 68040, so this is all you need to know about it. ◇

In two cases, each Macintosh Quadra's ROM helps with cache flushing. First, the ROM flushes the data cache after it loads a resource into memory, moves a heap block, or creates a jump table. This ensures that the data in the affected areas are not “stale.”

Second, in a situation in which an alternate NuBus bus master alters memory outside the control of the 68040, the Macintosh Quadra ROM communicates with it only through pages of memory marked as uncacheable.

- *New exception handlers:* The 68040 handles exceptions by using a new method called *instruction restart*. This means that it can abort instructions that are partially through the integer pipeline. The 68040 also creates its own exception stack frames for its access errors (which are the 68040's equivalent of bus errors). Each Macintosh Quadra's ROM correctly patches the startup code and the bus error handlers for the Slot and Memory Managers.

- *Speeding up floating-point computations:* In the Motorola 68040 processor, the math coprocessor (formerly on a separate chip, the 68882) is in the processor itself. This means that most routine math operations execute very quickly. This built-in coprocessor doesn't execute all the 68882's instructions, however, so each Macintosh Quadra's ROM emulates these instructions.

The floating-point instructions not handled are the ones that Motorola found were least often used, including trigonometric (for example, sine, arcsine, and hyperbolic sine), logarithmic, and exponential functions.

Since these emulations are in software, they may execute more slowly than they do in the 68882 hardware. You may wish to examine the code your compiler produces and either avoid an emulated instruction or patch the code to jump to a custom library routine (which will execute faster).

- *The 68040 MOVE16 bug*: The 68040 includes a new and useful instruction, MOVE16, that moves 16 bytes of memory, starting at a multiple-of-16-aligned address. Unfortunately, the current implementation of the 68040 may not work properly if your program calls MOVE16 when results from previous instructions have not been written to memory or cache.

One solution is to precede the MOVE16 with a NOP instruction, which (believe it or not) actually does something—it forces pending writes to finish. A better solution is to use the system-software routine BlockMove. This is a general-purpose, highly optimized data-movement routine that uses MOVE16 when it's appropriate and advantageous to do so.

- *Byte smearing and video cards*: The 68030 and earlier 680x0 processors do something called *byte smearing*, in which a datum that is being moved in a single-byte transfer “smears” to an adjacent byte (for more detail, see Macintosh Technical Note #282, “Smear Tactics”). This smearing behavior was never officially part of the processor's behavior, and Motorola removed it from the 68040.

Some third-party video cards use byte smearing and will require changes to work correctly on the two Quadra models. Early Apple video cards—the Macintosh II, Portrait, and Two-Page Video Cards—also do this; for these cards, Macintosh Quadra system software automatically patches the code that needs to be changed.

MACINTOSH CLASSIC II

Is it possible for more than one Macintosh to be a Classic? Of course, different people have different needs and the Macintosh Classic II defines the “next step up” for users. Compared with the original Macintosh Classic, this computer delivers more elbowroom (in the form of larger hard drives, a larger memory capacity, sound-input capability, and an optional 68882 math coprocessor). It also delivers more power, in the form of a Motorola 68030 processor running at 16 MHz. There's also an attractive \$699 kit that allows Macintosh Classic owners to upgrade to a Classic II.

Macintosh Classic II

Microprocessor

- MC68030, running at 16 MHz (2.5 times faster than a Macintosh Classic)
- Optional 68882 math coprocessor (and 3 MB ROM) can be added

Interfaces

- One ADB port
- Two serial (RS-232/RS-422) ports
- SCSI interface
- Port for external SuperDrive
- Monaural sound-in and sound-out ports

Video display

- Built-in 9-inch diagonal monochrome display, 512 by 342 pixels

Disk drives

- Internal 40- or 80-MB SCSI hard disk
- Internal 1.4-MB SuperDrive

Memory

- 2 MB RAM (main memory) on the logic board, 2 SIMM slots
- Expand to 4, 6, or 10 MB RAM total
- 512K of ROM

Other

- Uses System 7.0.1
- ADB keyboard (with keypad) and mouse
- Microphone
- Line voltage: 120 volts AC, 47 to 63 Hz; 100 watts max.
- Weight: 16 to 17.1 lb. (7.3 to 7.8 kg), depending on configuration

U. S. Suggested List Pr.

- With 2 MB RAM, 40-MB hard disk—\$1,899
- With 4 MB RAM, 80-MB hard disk—\$2,399
- Upgrade kit for Macintosh Classic (only)—\$699

Table 3: Fact sheet for the Macintosh Classic II

Classic + 68030 + More. The simplest description of the Macintosh Classic II is to say that it's a monochrome, all-in-one-box 68030 Macintosh that has about the same power as a Macintosh LC.

Although the Macintosh LC and the Macintosh Classic II are very different physically, the Classic II was designed around the architecture of the Macintosh LC. Table 3 lists the main features of the Macintosh Classic II, but here are the features that distinguish it from the original Macintosh Classic and the Macintosh LC:

- *68030:* The Classic II uses a Motorola 68030 running at 16 MHz (the same speed as the Macintosh SE/30, which will eventually be discontinued); the original Macintosh Classic uses a 68000 running at 8 MHz. Because of the 68030's internal memory-management unit (MMU), the Classic II can run System 7's Virtual Memory (the original Classic can run all the System 7 features except VM).

Because of the 68030 and other factors, the Macintosh Classic II runs approximately two times as fast as a Macintosh Classic.

- *Math coprocessor and ROM expansion:* Unlike with the original Classic, you can internally add a Motorola 68882 math coprocessor and 3 megabytes of ROM to the Classic II. (Developers may or may not find a use for the extra ROM, but the 68882 expansion option is definitely useful.)

- *More memory:* The Macintosh Classic II comes with either 2 or 4 megabytes of memory and System Software 7.0.1 (the original Classic came with 1 or 2 megabytes of memory). Users can expand the Classic II to a maximum of 10 megabytes, which contrasts with the Classic's 4-megabyte maximum.

- *No expansion slot:* Unlike the Macintosh LC and SE/30, the Classic II does not have an internal expansion slot for add-in hardware cards that provide such things as support for an external color or large-screen monitor.

- *Sound input:* Like the Macintosh LC—and unlike the original Classic—the Macintosh Classic II can record sounds. Of course, all Macintosh models can play sound back.

- *The Eagle custom gate array:* This new chip is called “the heart of the hardware design” because it integrates the functions of several chips into one.

It does both 24- and 32-bit address mapping and includes timing, video and clock generation, sound, and other “glue” functions.

Compatibility Issues. The Macintosh Classic II has the same compatibility concerns as similar Macintosh models. See the first four points of “Compatibility Issues” in the PowerBook section for details.

LaserWriter IIx

Marking engine

- Canon LBP-SX; lifetime minimum 300,000 pages

Processors

- 20 MHz 68030, ASICs, and I/O processors

Imaging languages

- Adobe PostScript Level 2; emulates HP PCL4+
- Supports all TrueType and PostScript fonts

Memory

- 2 MB printer memory, expand-able to 32 MB
- 2 MB ROM

Other features

- Eight pages per minute maximum
- 11 fonts built-in
- FinePrint enhances print quality
- PhotoGrade (needs 5 MB printer memory) improves scanned images
- Simultaneous connection (“all ports active”) to LocalTalk, RS-232
- SCSI port for hard disk (used for font storage)

U.S. Suggested Retail Price

- \$3,599 (\$1,549 for LaserWriter IIx upgrade kit, less trade-in)

LaserWriter IIg

Marking engine

- Same as LaserWriter IIx

Processors

- 25 MHz 68030, ASICs, and I/O processors

Imaging languages

- Same as LaserWriter IIx

Memory

- 5 MB printer memory, expandable to 32 MB
- 2 MB ROM

Other features

- Same as LaserWriter IIx, except PhotoGrade is built-in and LaserWriter IIg also includes an Ethernet port

U.S. Suggested Retail Price

- \$4,599 (\$2,549 for LaserWriter IIg upgrade kit, less trade-in)

Table 4: Fact sheet for the LaserWriter IIx and IIg

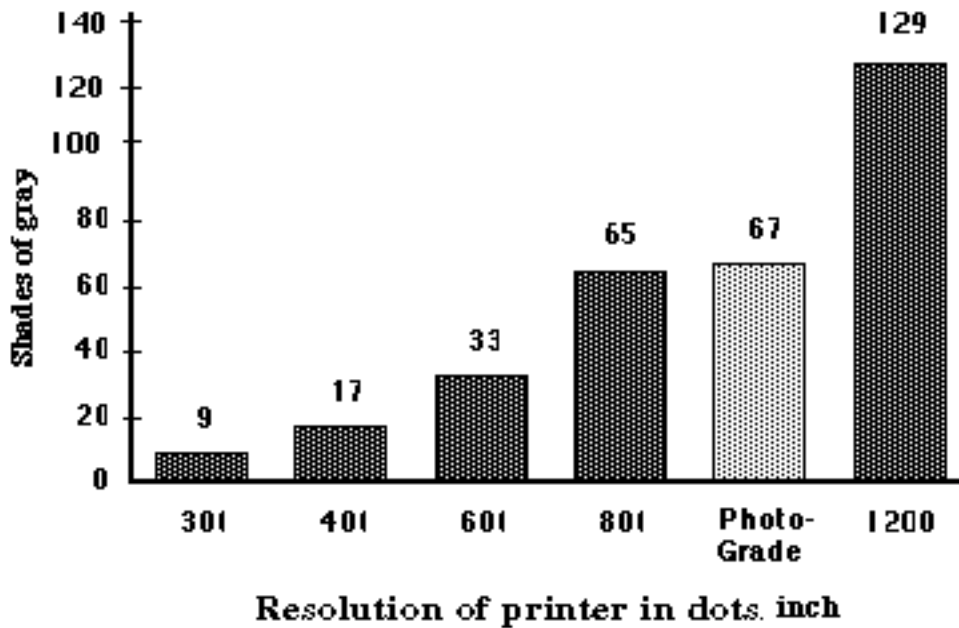


Figure 2: *PhotoGrade and gray scales—the LaserWriter IIf and IIfg can produce about the same number of shades of gray as a conventional 800-dpi printer, even though they are 300-dpi printers.* Figure 2: *PhotoGrade and gray scales—the LaserWriter IIf and IIfg can produce about the same number of shades of gray as a conventional 800-dpi printer, even though they are 300-dpi printers.*

LASERWRITER IIF AND IIFG

One interesting thing about magic is that it allows you to break the rules. For example, the rules say that to get a more detailed image from a laser printer, you have to go to a more expensive laser print engine with a higher resolution—and by the way, the printer will need a lot more memory, your image files will be much larger, and they will take much longer to print. To that, magic says, *It ain't necessarily so.*

The magic that Apple engineers put into the LaserWriter IIf and IIfg has two names: *PhotoGrade*, which makes scanned images appear much sharper than they do on other laser printers that print 300 dpi (dots per inch), and *FinePrint*, a feature that smooths the contours of print and line art. (There's more magic in the fact that these LaserWriters know what to treat which way.)

And most of us would agree that there's some magic in two printers that work at up to double the imaging speed of their predecessor (the LaserWriter IINTX) while costing less.

(To clarify imaging versus printing: The LaserWriter IIf, IIfg, and IINTX all have a top output speed of eight pages per minute, but the LaserWriter IIf and IIfg can image a complex document up to twice as fast as can the LaserWriter IINTX. *Imaging* is the process by which the document file is “drawn” into internal printer memory; its result is a bitmap image that the laser print engine needs to do the actual printing.)

What's the bottom line of all this? With the pricing and features of the LaserWriter (along with the new Apple OneScanner, discussed below), users will be creating media-rich documents that will make better-looking graphics an integral part of how people communicate with each other.

Table 4 summarizes information about the LaserWriter IIf and IIfg. Both printers are identical except for their processor speed, the amount of memory each

comes with (both can be expanded to a maximum of 32 megabytes), their input ports (the LaserWriter IIg has an Ethernet port), and their price.

Making Them Faster. The easiest way to make a printer faster is to upgrade its processor and memory. The LaserWriter IIc and IIg each exemplify this by using a 68030 processor running at 20 and 25 MHz, respectively, with a maximum of 32 megabytes of printer RAM. (The LaserWriter IIINTX uses a 68020 running at 16 MHz, with as much as 12 megabytes.)

Memory becomes important when you're dealing with complex graphics—the printer can save time by caching (storing) font definitions in its memory rather than calculating them for each document.

Like other Apple products (most notably the Macintosh IIx), these printers do not get their speed from improving only the processor. A faster computer (and Apple LaserWriters *are* computers) can be held back by bottlenecks elsewhere in its design.

That's why Apple believes in what it calls "balanced" enhancement—that is, improving all the components needed to maximize the overall improvement.

Like the Macintosh IIx, the LaserWriter IIc and IIg contain several IOPs (Input/Output Processors) that take care of specific tasks that would otherwise be handled by the 68030 processor.

In addition, one ASIC (application-specific integrated circuit) takes over when the processor has finished translating a page's description into an actual bitmap. This ASIC takes over the time-consuming task of "feeding" the bitmap image to the laser print engine, freeing the 68030 to use its resources more productively.

Another source of speed in the LaserWriter IIc and IIg is the printers' use of Adobe PostScript Level 2. This extension to the original PostScript page-description language certainly adds more features, and it also executes many common PostScript commands faster than before.

One of the few features unique to the LaserWriter IIg is the Ethernet port that allows it to connect to an Ethernet network. A LaserWriter IIg that uses its Ethernet port can run more than twice as fast as it would if it were using its LocalTalk port.

Making Them Look Better. Two of the most important technologies inside the LaserWriter IIx and IIg are PhotoGrade, for creating high-quality gray-scale images, and FinePrint, which smooths the contours of printed text and line art. (Remember that PhotoGrade exists within the LaserWriter IIx but is inactive until the user increases the printer's memory to 5 MB or more.)

Of the two, PhotoGrade produces the more dramatic improvement in print quality. PhotoGrade makes it possible for the same laser print engine to print more shades of gray (67, as opposed to 9 in previous LaserWriters—see figure 2). It does this by storing each point in the image as a 4-bit gray-scale pixel instead of the usual 1-bit monochrome pixel.

The LaserWriter IIx and IIg implement this improvement by being able to horizontally lengthen or shorten the pixels that the laser print engine makes. In practice, the 16 possible values available to a 4-bit pixel translate (with current laser engines) to 8 visually different shades of gray.

By using these various shades of gray along with PostScript halftoning, these 300-dpi printers are able to represent as many levels of gray as does an 800-dpi printer, which requires almost double the amount of printer memory of the LaserWriter IIg.

On the other hand, FinePrint uses dedicated hardware (a custom ASIC) to smooth the black/white boundary of text or line (1-bit) art. It does this by elongating dots horizontally to fill in the areas that normally create the “stair step” look of coarse bit-mapped graphics. The Laser-Writer IIx and IIg keep track of which graphics are mono-chrome and which are gray scale and apply FinePrint and PhotoGrade, respectively, to them.

With line art and most sizes of type, FinePrint output looks very similar to that of higher-resolution printers. (The effect is subtle, showing up best when you look at print with a magnifying glass.)

Using Fine-Print gives a smoothness approaching 2,400 dpi horizontally without increasing the image's file size, the time it takes to print the image, and the amount of memory the printer needs.

Note that this smoothness is not the same as higher resolution. These LaserWriters don't print, say, 4-point text as well as an 800-dpi printer would, but they do print larger text (certainly 9-point characters and up) that looks much smoother than it would on other 300-dpi laser printers.

Making Them More Useful. The LaserWriter IIx and IIg make it easier for you to use them in a mixed (Macintosh and IBM DOS) work environment. On the Macintosh side, you can connect to these printers through the LocalTalk connector (or, with the LaserWriter IIg only, the Ethernet port). On the IBM side, you can connect through the serial (RS-232/RS-422) or Ethernet ports.

Because of a feature called *port arbitration* or “all-ports-active,” these LaserWriters print jobs from any port on a first-come-first-served basis. (With previous LaserWriter models, you had to tell them which one port to use, and they would accept input from that port only.)

In a mixed work environment, a customer might think of buying a LaserWriter IIx or IIg instead of separate printers for Macintosh and DOS computers. Also, since these two printers emulate Hewlett-Packard’s PCL 4+ (a popular page-control language meant for DOS computers and the HP LaserJet IIP printer), the majority of DOS programs can print on these LaserWriters without problems.

Apple has also made the LaserWriter IIx and IIg configurable through the new Laser-Writer utility program that ships with them, LW Utility 7.1. This program makes it easy to set up and manipulate these printers—for example, you can change serial parameters in software via a dialog box (rather than having to change a cryptic combination of DIP switches on the printer).

Making Them Count. As always, there are things that you, the developer, need to know. First and most importantly, make sure your program is versatile in its use of graphics. Your customers will be putting far more graphics into their documents than they do now. They will gravitate to programs that don’t just let them use graphics—all programs do that. They will gravitate to programs that make it easy for them to use more graphics in more-powerful ways.

You should get to know Adobe PostScript Level 2; its new features may suggest enhancements you can make to your programs. PostScript Level 2 includes many new features that extend what your program can do with print—such things as improved support for patterns, device-independent color, forms (arbitrary graphic elements meant to be reused), and new graphic manipulations.

The definitive book on the subject is *PostScript Language Reference Manual, Second Edition*, by Adobe Systems, Inc. (Addison-Wesley, 1990). For information on getting the PostScript Software Development Kit (which includes comprehensive documentation and software), call Adobe at (800) 833-6687.

The other advice you need is an old friend: “*Don’t assume...*” Don’t assume how many grays the printer will print or how many lines per inch are in the halftone screen. In general, the simplest recipe for printing is to send all the image information you have and let the LaserWriter II_f or II_g print the best image it can.

PhotoGrade is optimized to provide the best output with the default halftone screen and transfer function. If your program sets these itself, you may not get the best grays.

You can also have your program locate and find the text-only PPD (PostScript Printer Description) file for the printer and get the current printer’s model from the PPD file. Your program can’t query the LaserWriter to find out its characteristics.

(Communication is one-way from the Printing Manager to the printer. It has to be this way—or else how would you handle spooled documents or documents to be printed later?)

ONESCANNER, OFOTO, AND HYPERSCAN 2.0

When was the last time you scanned an image into your computer? Yeah, you wanted to add a photograph to one of your reports but you just didn’t have time to mess with the scanner—and you know how you can never get an image exactly right, no matter how long you fool with it. Sound familiar? Well, Apple hopes to eliminate the hassle of scanning with OneScanner and Ofoto.

Apple’s new One-Scanner flatbed gray-scale scanner and its associated software, Ofoto and (for HyperCard 2.0 users) HyperScan 2.0, offer users more in two important areas: ease of use and value (performance for a given price).

Not content to deliver just an improved piece of hardware for less money, Apple decided to look at how people use scanners and streamline that as much as possible. Table 5 lists the features of One-Scanner and its software.

The One-Scanner hardware is simple enough, though. It recognizes 256 gray-scale levels (the original Apple Scanner recognizes 16). It connects to your computer as a SCSI peripheral—and it does it for less money (the U.S. suggested retail price of the OneScanner is \$1,299, \$500 less than that of the now-discontinued Apple Scanner).

Ofoto? Oh My! Ofoto is (for me, at least) one of those programs that feels so *right* when you use it that you wonder how you could ever have tolerated previous products. You put the document into the scanner, choose the output device for which the scan is being done, and press the Autoscan button—and that's all you have to do.

Ofoto then (depending on what's being scanned) does a quick prescan of the scanner's image area to find out where the image is, fully scans (at the appropriate resolution and bit depth) the small rectangle of content that includes all the nonwhite graphics it sees, resizes the resulting image, straightens it, and crops it according to its best guess at what the image's actual content is. The result, for many of the things most people want to scan in, is exactly what the person using the scanner wants, in one try and without frustration.

In some cases, you want to manipulate the image further, perhaps recropping or resizing it, or you may want to specify how the image is to be scanned. (With Autoscan, Ofoto analyzes the image and makes these decisions itself; you may find that you want to change them.)

The great thing about Ofoto is that now that it has the scanned image in memory, it performs whatever manipulations you want from the image in memory *without needing to rescan the original* for every change you want to make. (This feature alone takes most of the grief out of scanning.)

Apple OneScanner

Ofoto scanning software

- Autoscan—Ofoto automatically examines image, adjusts scanning parameters, straightens and crops final image
- Adaptive calibration delivers the best possible image for a given printer or output method
- Selection and scaling tools for enlarging or reducing an image
- Virtual imaging system uses disk space to handle images too large for scanner's memory
- High-quality rotation and scaling for image manipulation without loss of quality
- Supports PICT, TIFF, EPS, and MacPaint files

Hardware

- Type: Flatbed; scan area of 8.5 x 14 in.
- Resolution: 72 to 300 dots per inch (dpi)
- Gray-scale levels: 256 levels (up to 8 bits/pixel)
- Scaling: unlimited
- Interface to computer: SCSI port

Other

- HyperScan 2.0 makes use of HyperCard 2.0 features
- Requires Macintosh System 6.0.7 or later

U. S. Suggested Retail Price

\$1,299

Table 5: Fact sheet for the OneScanner and its software

What's the most annoying thing about scanning? Right—not being able to get the #*\$%@\$ image exactly straight in the scanner, so you always have almost vertical and almost horizontal lines that show a maddening slight stair-step effect. The auto-straighten feature of Ofoto looks for a horizontal line near the top of the image and—in memory—rotates the image to make that line exactly horizontal! (There's a Straighten menu item you can use if you're scanning manually.) What a time-saver that is!

So what's the next-most-annoying thing about scanning? Right—scanning and rescanning the image until—after several frustrating cycles of scan, measure, adjust—you finally scale the image to fit the 3-inch-wide hole waiting for it in your document.

In Ofoto, you just display the Resize dialog box and enter a width of three inches; Ofoto delivers a resized image exactly three inches wide. You can similarly specify an exact height or scaling factor and get back exactly the image you want.

And Ofoto automatically scans the image at whatever resolution is needed to get the resolution you want—or the closest it can get—for the final output device; you don't have to worry (or know) about such details.

(OK, truth-in-advertising time: If you're going to a really-high-resolution output device such as a Linotype and want the absolutely best image possible, you may want to rescan the image once you've gotten it to the right scale. But for

output devices such as laser, inkjet, and dot-matrix printers—360 dpi and less—you don't need to rescan.)

Ofoto offers on-screen tools—e.g., a selection rectangle, lasso, pencil, eraser, and magnifier—and several image-manipulation commands—such as Invert, Sharpen, Flip Horizontal, Flip Vertical, and Rotate. And, of course, you can manually override any parameter Ofoto chooses for you.

Resisting the Need to Calibrate. And now we get to the pièce de résistance of Ofoto, its philosophy of auto-calibration.

Again, Apple engineers looked at the process of scanning and saw it from a new perspective, considering not a scanned-image file on-disk as the final product, but rather the image in its final form—printed on bond paper, newsprint, or fax paper, or displayed on-screen as a monochrome image. Obviously, the same photo looks remarkably different on each of these media, but nobody ever thought of calibrating a scanner to optimize for the specific medium.

The method used to calibrate Ofoto for a given medium (if it isn't in the long list of calibrations already provided by Apple) is breathtaking in its elegance and simplicity.

First, the user directs Ofoto to print out a test page of gray scales, from white to black. The user gets that printed image to the final medium (by, for example, transmitting it as a fax) and then lets Ofoto scan in the resulting image and calculate how best to adjust the image for that medium. Of course—why didn't anyone think of that before?

The Final Result. So look what happens with Ofoto: The user scans the image once and gets final output that is better than what most users would have gotten after a dozen scans and an hour of wasted time.

What does this mean to you, the developer? With both the OneScanner and the LaserWriter II^f and II^g, once users have bought these two products, they will be putting a lot of images into their documents.

Be like Ofoto; don't just allow them to paste the images into your program's documents—make it easy for them, give them creative choices, help them be more productive. By doing so, you give your product an edge over your competitors', and both you and your customer win.

APPLETALK REMOTE ACCESS

Remember VisiCalc? It was the world's first spreadsheet program, and that program alone induced many otherwise-conservative office workers to buy Apple IIs. (And this was back in 1980, when personal computers were little more than a gleam in the fanatic hobbyist's eye.) What a revolutionary product VisiCalc was—people bought thousands of dollars' worth of computer systems just to run that one program!

Granted, most of us don't need an excuse to buy Macintosh computers. But for anyone who does, AppleTalk Remote Access is definitely the excuse to do so. With a modest amount of hardware (see table 6 for details), you can be anywhere in the world and access not just your other Macintosh (at home or at work) but also the entire AppleTalk network connected to it. In other words, you can use a 2,400-bps or higher-speed modem and a telephone line to become a remote node to whatever AppleTalk network is on the other end of an ordinary telephone line.

If you're not convinced yet (and *still* need an excuse to buy a Macintosh PowerBook), here's another reason: AppleTalk Remote Access, a \$200 value, ships free with every Macintosh PowerBook. Yikes! For \$3,000 to \$5,000, depending on your taste in PowerBooks, you can sit in a hotel room (or at home) and do almost anything you could do on your Macintosh at work—and all this on a computer light enough to take with you! For me, AppleTalk Remote Access is the VisiCalc of 1991.

How It's Different. There's nothing on the market quite like AppleTalk Remote Access, so it's important to point out its distinctive features:

- *It's easy to set up.* Many existing programs require a knowledgeable administrator to get things running. As with its other products, Apple designed AppleTalk Remote Access to be very easy to install—and equally easy to use.
- *It's symmetrical.* AppleTalk Remote Access acts as a bridge/router (for incoming calls) and a client program (for outgoing calls). Most other products handle only one of these functions, necessitating a second purchase to provide its inverse.
- *It's secure.* The dream of accessing your network when you're somewhere else is also a nightmare about some unauthorized person doing so too. AppleTalk Remote Access has several levels of security. First and simplest, you

can require callers to supply a password before they can connect to the remote computer and network.

Second, you can specify that callers can access the remote Macintosh but not the network to which it's connected. Third, you can make the remote Macintosh do a callback (call a specified telephone number before giving access to a user). This ensures that the user has both the correct password and access to a physical location known to be secure.

- *It's a real node on the network.* Some programs allow you to view and control the Macintosh on the other end of the telephone line, and they use the telephone connection to keep the screen of the local Macintosh "painted" with what's actually happening inside the remote Macintosh. AppleTalk Remote Access, on the other hand, makes the local Macintosh an actual node on the network.

- *It's an "individual" solution.* AppleTalk Remote Access has been described as a "personal service" program, meaning that it's meant for individual users. Apple also has plans for multiuser solutions.

Developer Opportunities. If your company sells 2,400-bps and faster modems, you'll definitely want to make sure your modem works with AppleTalk Remote Access (several dozen modems already do).

You will need to supply a CCL (connection-control language) script to be packaged with your modem; alternatively, you can send your finished CCL script to Apple (via AppleLink at the address APPLETALK) and we will distribute it to the major on-line services. To learn how to create your own CCL scripts, you can purchase the AppleTalk Remote Access Modem Toolkit from APDA (see "Now Available from Apple" on page 2).

For remote access to become successful in the Macintosh environment, Apple and third-party developers such as you need to agree on a protocol for dial-in access. Apple has been working with third-party developers to create such a standard, tentatively called ARAP (AppleTalk Remote Access Protocol). If you want to get involved in this process, you should send an AppleLink message to the APPLETALK address mentioned above.

One of the greatest developer opportunities is to provide a product that can service multiple phone lines, using the ARAP protocol. This could be Macintosh software that allows one Macintosh to connect multiple users (on multiple

phone lines) to the network, or it could be a stand-alone hardware product that connects directly to an AppleTalk network.

To remain competitive against other platforms' environments, the Macintosh environment has to provide services and features that are unavailable elsewhere. We believe that simple but powerful remote access is such a service, and we need your support to make this commonplace. And of course, our success will be built on top of yours.

APPLESHARE SERVER 3.0

The very existence of AppleShare Server 3.0 is proof that plenty of people are using Macintosh personal computers and that they're not using them alone. Groups of up to 120 simultaneous users—much larger than the “small workgroups” of 50 for which AppleShare was created—are wanting to stay connected to each other. AppleShare Server 3.0 (see table 7) broadens the “highway” on which these users are depending. It also opens AppleShare to third-party extensions, which may mean new customers for you and your products.

AppleTalk Remote Access

Function

- Allows a Macintosh with a 2400 baud or faster modem to access a remote AppleTalk network

Requirements

- Any Macintosh running System 7, with 2MB or more (recommended: 4MB or more), and selected 2400 baud or faster modem

Security features (selected list)

- Only authorized users can connect
- Keeps an activity log of the 300 most recent interactions it has had
- Callback option causes AppleTalk Remote Access to call the user at a preset telephone number
- Receiving Macintosh can limit access to itself only (not network)

U. S. Suggested Retail Price

- \$199; bundled free with PowerBook 140, 170, and 190

Table 6: Fact sheet for AppleTalk Remote Access

System 7 Makes It Happen. That the AppleShare Server 3.0 file and print servers are implemented as standard System 7 programs has several implications, all of them good. Because the file and print servers are both “well behaved” System 7 programs, the Macintosh on which they’re running (called the server machine) can also run other System 7 programs. (Until now, AppleShare required a “dedicated” Macintosh that could do little more than be an AppleShare server machine.)

Granted, you don’t want to run a CAD or high-end graphics program on the server machine, but AppleShare Server 3.0 does free the server machine to do some other tasks (perhaps optimizing the use of a dedicated Macintosh to control several network-related services).

Previous versions of Apple-Share shared the server machine only with what were called AppleShare foreground applications, which could access files only within the server machine’s Server Folder. AppleShare Server 3.0 removes such limits: System 7-friendly programs on the server machine run without restriction.

Opening AppleShare to Third-Party Programs. AppleShare Server 3.0 makes it possible for you, as a developer, to add AppleShare-related functions to your programs through a standardized set of application program interfaces. By doing so, Apple hopes to make available to users many specialized server functions that Apple cannot provide.

Server control calls allow you to monitor and control file-sharing and -serving functions. This includes such things as getting the status of users and volumes, starting or shutting down the server, and disconnecting users. The server event mechanism allows programs to selectively monitor (but not modify) server-related events. This will allow your program to do such things as compiling statistics on server use and debugging server-related problems. Used with server control calls, they can provide numerous support functions that users want.

Apple encourages you to put networking capabilities into your programs. To get more information on how to do this, send a message to AppleLink address APPLELINK with a description of your program and what server capabilities you'd like to add to it.

AppleShare Server 3.0 is important because it gives its users enough elbowroom to do things that were difficult or impossible to do previously. Apple also hopes that opening the AppleShare architecture will enrich the Macintosh environment with new network and file-sharing tools and applications.

“RAISING THE BAR”

“Raising the bar” is a favorite phrase at Apple these days. It basically says, “See here, we’ve added this new feature and we think that people will like it so much that from now on, they won’t accept anything less. To get people’s attention, our competitors will have to add both this and more.” Raising the bar makes people see your product as the front-runner and your competition’s as being “me too” and behind the cutting edge.

These 12 new products certainly raise the bar—not just for the competition but also for Apple itself and for you. Macintosh users will expect more from you—better versions of existing products and new products that will leave us slack-jawed in amazement as we beg for the privilege of buying them.

Apple’s October 1991 products raise the bar for Macintosh computing. Now it’s your turn.

AppleShare Server 3.0

Function

- Allows a network of up to 120 simultaneous users to share files and resources
- Implemented as three independent programs: AppleShare File Server, AppleShare Print Server, and AppleShare Admin
- Can spool up to 5 AppleTalk printers

Requirements

Any Macintosh with 4 megabytes of memory (RAM) and System 7

Maximum number of:

- Simultaneous users: 120
- Open files: 346
- Shared volumes: 50
- Apple II network startups: 40
- Users + Groups: 8192 (recommended: up to 2000 total)

Security features (selected)

- Password aging
- Ability to disallow saved passwords
- Two-way scrambled user/server authentication
- Temporary accounts with set expiration dates
- Logoff of user by administrator
- Folders default to exclude guest access

Administrator functions (selected list)

- Can control number of connected users
- Can send messages to any user
- Can change CD-ROMs without bringing the server down

Sharing privileges

- 3 levels of folder access, 5 levels of access to a folder's content
- Allows users to allow access to a folder only
- A folder can inherit the privileges of the folder containing it
- User can give access privileges to a single user

Other

- Can work with multilaunch applications to limit launches (to enforce site license agreements)
- Uses AFP (AppleTalk Filing Protocol) 2.1

U. S. Suggested Retail Price

- AppleShare Server 3.0: \$1,199
- For registered owners of AppleShare 2.0 file server: \$299
- For registered owners of AppleShare 2.0 file and print servers: \$199

Table 7: *Fact sheet for AppleShare Server 3.0*

A New Display, Too

One late arrival to this month's list of new products is the Macintosh 21" Color Display. This is Apple's two-page color monitor, which can display 1,152 X 870 pixels at 79 dpi, using 256 colors (from a palette of more than 16 million colors).

It has a 75-Hz screen-refresh rate (to eliminate flickering) and an auto-degaussing circuit. (This circuit degausses the monitor whenever you tilt or rotate it—a feature that is needed in a monitor this sensitive to ensure its color fidelity).

It also has a built-in antiglare screen, a tilt-and-swivel base, and three ADB connectors on its front (to help minimize cable clutter).

The U.S. SRP is \$4,599 and it will begin shipping in the U.S. in January 1992.

Thanks go to Adam Samuels for the Herculean task of putting the fact sheets in this article together. Thanks also go to Aneesh Shrikhande for the (alas, temporary!) loan of a PowerBook 100 and 140.

Apple Launches Antipiracy Plan

Joins SPA Protection Fund

Apple has long addressed the technical needs of its third-party developers and is now focusing increased attention on developers' business needs as well. Recognizing piracy as one of the issues directly affecting the bottom line of developers, the Apple Developer Group is committing significant resources to an antipiracy campaign.

At last month's Software Publishers Association (SPA) conference, in Orlando, Florida, Apple announced that it would

- join the SPA Copyright Protection Fund,
- sponsor French and Spanish translation and distribution of the SPA's videotape "It's Just Not Worth the Risk," and
- sponsor further development and distribution of the SPA's SPAudit software for the Macintosh.

At the time of the announcement, Kirk Loevner, director of the Apple Developer Group, donated \$30,000 to the SPA Copyright Protection Fund.

These activities follow Loevner's announcement at last August's Macworld Expo that Apple would undertake a three-pronged approach to antipiracy by collaborating with industry associations that are working to fight software piracy; using its position in the industry to raise awareness about software piracy among resellers, business partners, and user groups; and investigating potential technologies and other security strategies for developer implementation.

As a member of the SPA Copyright Protection Fund, Apple now has a voice in determining how the funds will be distributed to best educate computer users about piracy and can act in an advisory manner on litigation issues.

Apple is sponsoring the translation of the SPA videotape into Spanish and French and distributing it because software piracy is an industrywide issue, particularly in markets outside the U.S. Translating the tape makes it possible for an excellent educational resource tool to reach many more groups of people than an English-only version would.

The translated videotapes will be available in October through the SPA. Additionally, Apple will distribute them through its European offices.

Apple is also sponsoring the Macintosh conversion of SPAudit, a software-management tool that enables computers users to inventory their software.

Users can then take SPAudit's printout and compare it with their specific site-license list and/or software-purchase list to ensure that they are in compliance with software licenses. The Mac version will be available by the end of 1991.

Although the SPA is the only organization with which Apple has made publicly-announced plans at this time, Apple is also in contact with many other organizations, such as the Business Software Alliance (BSA), in Europe, and the JPISA, which is based in Japan.

Additionally, Apple has mailed SPA materials to all Apple user groups and is looking at even more extensive distribution of existing materials (such as the SPA videotape, brochures, and SPAudit) as well as the development and distribution of Apple antipiracy collateral.

There's a large team within the Apple Developer Group that's determining how Apple can best contribute to the fight against software piracy worldwide. Watch *Apple Direct* for reports of future developments. ◇

Wanted: QuickTime Apps, Info and Code Exchange

If you have a product that takes advantage of QuickTime, Apple wants to know about it. Tell us about it by sending a link to the AppleLink address COOL.APP.

We'll want to know how your product uses QuickTime and, of course, when you think your product will ship. Apple will consider these products for future marketing programs that involve QuickTime. (You can also use the COOL.APP address, incidentally, to inform Apple about any System 7-savvy application you might have—the address is closely monitored by Apple's Evangelism group.)

AppleLink is also the place to go to discuss QuickTime with other developers, exchange code, make feature requests, and debate the human interface for dynamic media. There's a new QuickTime discussion board on AppleLink called QuickTime Talk (Apple-Link path—Developer Support: QuickTime Talk), which contains both a code-exchange folder and a discussion board.

With the discussion board, we hope to accomplish two things: to solicit developer input on QuickTime and to encourage you to exchange your views and experience with other developers.

We'd like your opinions on topics such as

- the types of marketing and comarketing programs you would find most useful,
- the human-interface elements we've defined so far,
- the features you think would be most important for QuickTime 2.0,
- which QuickTime-related tools you need,
- and many others (see the QuickTime "Welcome to QuickTime Talk" document in the QuickTime Talk folder for more topics).

Besides the discussion board, you can also find QuickTime code and documentation on AppleLink (AppleLink path—Developer Support: Developer Services Bulletin Board: Developer Technical Support Folder: QuickTime Folder). ◇

Now Available from Apple

The following list shows which APDA products have become available to developers within the last several weeks. To get a full listing of all APDA products, check the current APDA Tools Catalog. For new product announcements and the most up-to-date price lists, check AppleLink (path—Developer Support: Developer Services:APDA).

If you're interested in the latest version numbers of all Apple system software products, check "Latest Rev" in the Information Resources folder on the current Developer CD. Latest Rev will also tell you where to obtain these system software products. In addition, the "Developer CD Highlights" section on page 3 of this issue tells you which new system software releases appear on the current CD.

A/UX Developer's Tools v.1.0, \$895
(Part # B0596LL/A)

A/UX Developer's Tools without MPW, v.1.0, \$395 (B0597LL/A)

AppleTalk Remote Access Developer's Toolkit, \$30 (R0128LL/A)

AppleTalk Remote Access Modem Toolkit, \$25 (R0129LL/A)

Getting Started in Macintosh C Programming (w/THINK C v. 5.0),
\$275
(B0473LL/B)

Getting Started in Macintosh Pascal Programming (w/ THINK Pascal
v. 4.0), \$275 (B0472LL/B)

Macintosh Development Tools and Languages Guide-book - 1992
Edition, \$6.95
or free with purchase of another product (A7Z2000/C)

MacsBug 6.2 with Discipline 2.0, \$15
(R0065LL/A)

MacsBug Reference and Debugging Guide, \$22.95 (R0063LL/A)

MacBug Reference and Debugging Guide with Software, \$34.95
(R0064LL/A)

MacTCP Developer's Kit
v.1.1, \$100
(M0704/C)

MacTCP Documentation Kit v.1.1, \$60
(M0217LL/B)

MacX25 Developer's Kit v.1.0.1, \$2,324
(B0314LL/C)

MacX25 Programming Library Kit, \$20
(M0788LL/C)

MacX25 Server Software Kit v.1.0.1, \$800
(M0711/B)

MacX25 Server Software Kit v.1.0.1 with Serial NB Card, \$2,124
(B0315LL/B)

ResEdit v. 2.1.1, \$29.95 (M0910LL/C)

ResEdit v.2.1.1 (disk only), \$15 (M0899LL/C)

Resorcerer, \$256 (T0437LL/A)

SADE v. 1.3, \$150 (M0026LL/E)

SADE v. 1.3 Update, \$50 (M0614LL/D)

THINK C v.5.0, \$210 (T0067LL/C)

THINK Pascal v.4.0, \$175 (T0068LL/D)

VISCA Driver v.1.0, \$50 (R0111LL/A)

XTND Developer's Kit, v. 1.3, \$30
(R0096LL/A)

APDA TOP TEN SELLERS*

1. DAL v. 1.3
2. E.T.O.
3. MPW v. 3.2 bundles
4. Developer University (includes all products)
5. Macintosh Common Lisp
6. MPW v. 3.2 Update bundles
7. MacApp v. 2.0 bundles
8. MPW C++ v. 3.1
9. MPW v. 3.2 Development Environment
10. Inside Macintosh, Volume VI

*as of 9/16/91

To place an APDA order from the U.S., contact APDA at (800) 282-2732. APDA's number in Canada is (800) 637-0029. And if you'd like to call the U.S. APDA office from outside the U.S., the number is (408) 562-3910. Additionally, if you're outside the U.S., you may prefer to work with your local APDA contact. For a list of non-U.S. APDA contacts, refer to the "International APDA Programs" page in the most recent APDA Tools Catalog.

System Software CD Source

Now you can get the latest versions of all of Apple's Macintosh system software—including non-U.S. versions—by simply booting your monthly Developer CD.

Each month, the CD will contain the latest release of all Macintosh system software, as of the day the CD content was frozen (typically about a month before you receive it). Watch the "Developer CD Highlights" column in each month's Apple Direct for a list of the system-software versions that have been updated since the previous month. ◇

Every Tog Has His Day

If you're a Bruce Tognazzini fan, you may have noticed that Tog likes to refer to past installments of his monthly Human Interface column. But if you haven't stored every issue of Apple Direct dating back to the beginning of time, or if you're in the habit of routing Apple Direct around the office, you may be out of luck when Tog tells you to look back at, say, the March 1989 issue for the pearls of wisdom he bestowed on developers at that time.

But don't despair—you can now access the "Collected Works of Tog" on AppleLink (Apple-Link path—Developer Support: Developer Services: Apple Publications: The Collected Works of Tog). This folder contains, in chronological order, every Apple Direct article Tog has written to date and will be updated each month. You'll also be able to find Tog's most recent columns on the monthly Developer CD, beginning next month, when we start archiving Apple Direct on the CD. ◇

Developer CD October Highlights

Each month this column will be your guide to the latest Developer Series CD, telling you what's new and notable. You may also want to scan the "What's new on this CD" folder, at the top level of the CD. Here you'll find aliases to every new and updated package on the CD, including all those listed below. On the October CD—Code Warrior, Vol. IX—you'll find the following highlights:

Quadra, PowerBook Data Sheets. Want to get the inside scoop on this October's hot new product announcements? Take a look at the Marketing folder. There you'll find data sheets for 12 new Apple products, including the Macintosh Quodras, the PowerBooks, the Apple OneScanner, and more.

International Versions of System 7.0. On the international scene, you'll find four new versions of System 7.0 software, including Danish, German, Swiss, and Turkish. Also take a look at the new glossaries of localized System 7 terminology for Dutch, Finnish, French, and German.

System 7 Hacks. As usual, we have numerous nifty new hacks. This month's hacks are designed to help you integrate System 7 into your new products. Ticket to Program illustrates the use of menus, windows, and alerts. Feature Teller demonstrates how to use the Gestalt and SysEnviorn managers.

CallFinder is an MPW tool that allows you to control the Finder through Apple events; this commando interface requires no knowledge or understanding of Apple events to operate.

Also take a look at Speed Control, a cdev that allows you to slow down any application; this handy tool is great for slowing down an animation so you can sync in sound.

Testing Tools. Code Warrior also features several new testing tools. VUAssist is a module that can be built in to a MacApp application to make it fully "testable" with Virtual User. Robix VI, a simple I/O window-based debugger, provides the bare-bones framework for a tailor-made debugger. The Apple II Monitor dcmd (xx) allows you to use MacsBug to resize and manage memory on an Apple IIe card.

Improved Search Tools. Look for enhanced search speed with our prototype of the Search Knowledge Base. The Developer Info Assistant 3.0 has also been updated to provide a seamless link between CustomLink and AppleLink, allowing you to send and receive links from DTS without manually launching AppleLink. ◇

New Dev. U. Course on Macintosh Debugging

Responding to developer requests, Apple's Developer University has expanded its curriculum with a new two-day course, "Introduction to Macintosh Debugging." The course offers practical instruction on the use of debugging tools for experienced Pascal and C programmers who are writing Macintosh software.

Students will be introduced to source-level debugging and will then take an in-depth look at machine-level debugging with MacsBug. Instruction will encompass the fundamentals underlying application development, including 68000 assembly language, memory constructs, stack operation, A-Traps, and identification of the elements of compiled code.

In order to maximize the effectiveness of the instruction, Developer University's course designers have built special educational tools to illustrate debugging techniques. The ClassBug applications for Pascal and C provide examples that can be run and, with the click of a button, viewed at the source- or object-code level. The examples contain bugs that students can flush out with the help of on-line notation and multiple levels of hints.

Another distinguishing aspect of the course is its emphasis on "anti-bugging," a set of coding conventions and stress-testing procedures that reduces errors and detects problems before they can crash the machine. This approach to anticipatory debugging is complemented by discussion of topical issues relating to System 7's virtual memory and 32-bit environment.

In addition to MacsBug and ClassBug, students will use the THINK C and THINK Pascal source-level debuggers and examine other tools such as SADE, TMON Professional, SourceBug, and The Debugger. Reference materials and a sourcecode library of test and debugging tools will be given to students to take home.

Prospective students should have at least six months of Pascal- or C-programming experience, have developed at least one Macintosh application that uses menus and windows, and understand the use of pointers and handles. Programmers at both the intermediate and advanced levels ought to be able to cut development time with the insights into Macintosh debugging that they gain.

For additional information on the new debugging course and other classes, contact the Developer University Registrar at (408) 974-6215 or, via Apple-Link, at DEVUNIV. ◇

DTA has Tech Info First

Where can you search for the most up-to-date technical information? Your first guess might be the Developer CD, and true enough, the now-monthly CD is a timely source for Q&As, Tech Notes, develop articles, and other searchable resources. But don't forget the Dev Tech Answers (DTA) library on AppleLink.

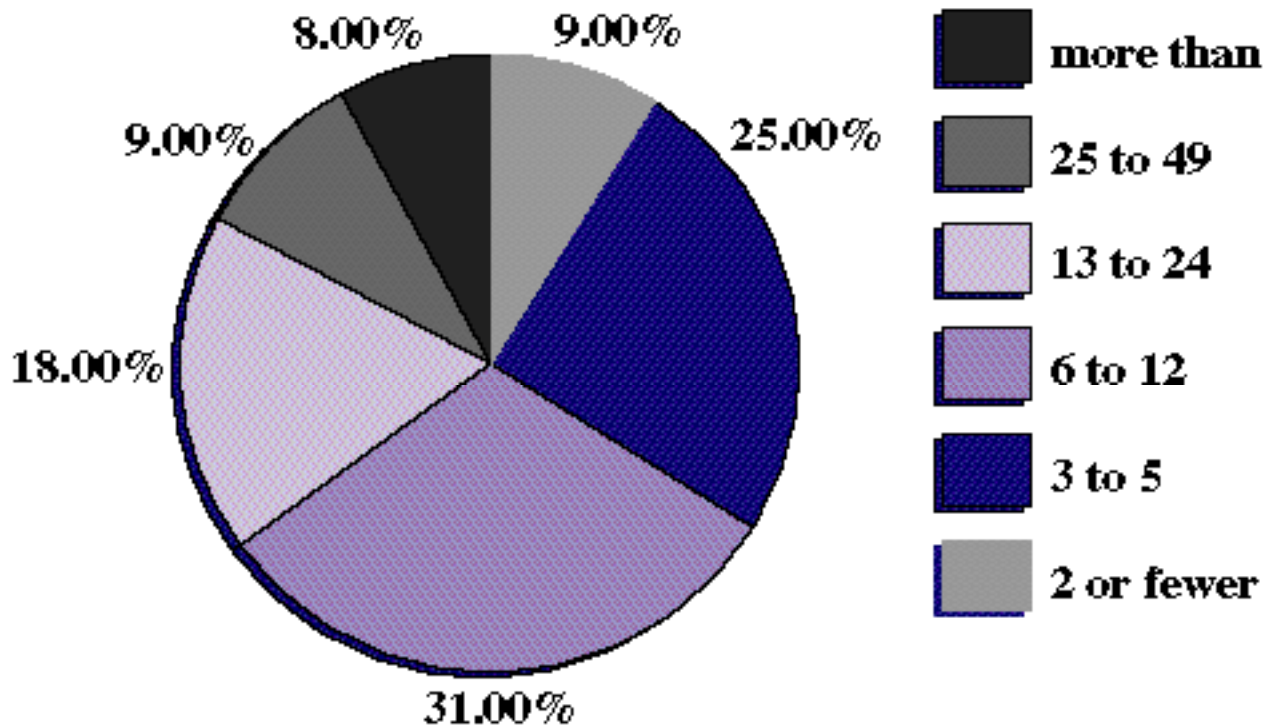
The DTA is now the place where you can find technical information before you can get it anywhere else. The contents of the latest CD Knowledgebase (which includes Tech Notes, Human Interface Notes, Inside Mac, develop articles, and hundreds of question-and-answer pairs pulled from the Developer Technical Support database) will get uploaded to the DTA at about the time the CD gets pressed, so AppleLink's instant distribution capabilities will get the information to you first.

Our recommendation is that you search the CD when you're looking for technical information and then log on to AppleLink to see if there's more-recent information that might be of use to you.

The DTA lets you do full-text Boolean searches. When you do a search, "hits" appear in a window where you can then see the full text of applicable questions and answers. Besides providing search capabilities, the DTA also organizes data in an indexed format. The AppleLink path is Developer Support:Dev Tech Answers. ♦

Size of Macintosh Development Teams

When the Apple Developer Group surveyed its U.S. commercial developers in December 1990, we asked how many employees each firm had dedicated to Macintosh development. As the chart shows, more than half the respondents had Macintosh development teams with more than two but fewer than 13 employees.



SOURCE: Apple Developer Group, 1990

Case Study: One-or-More Buttons

Dear Tog,

Have you ever come across a good control that gives users a “one or more but not none” selection? What I need would be something like a cross between radio buttons and checkboxes, allowing users to select as many options as they want but always keeping at least one selected.

And if such a control does not exist, how should it look? I would have no problem writing the control definition(s), but I need a good visual way of conveying to users how the control behaves.

—Neil Steiner
Highgate Cross & Cathey

Right after I got this letter, I contacted Neil and told him I had accepted his challenge. Not only is this just the sort of thing I like to do but I also thought I could use the project to demonstrate many of the design principles I've been propagating these many months. What I hadn't expected was that I would have to fight every inch of the way against the Big Three enemies of successful software design: rationalization, assumption, and denial.

Since I've been writing so much about them lately, I just assumed I had gained at least temporary immunity. “Assumed...” Yes, I think I see the problem.

Here are the guidelines for expanding the Macintosh interface from my December 1990 column, “The Evolving, Adaptive, Consistent Environment:”

- If it ain't broke real bad, don't fix it.
- Build on existing visual/behavioral language.
- Invent new objects, with new appearances, for new behaviors.
- When possible, evolve objects rather than starting from scratch.
- Make changes clearly visible.
- Interpret users' responses consistently.
- Multiplex meanings.

In addition to these, I also intended to follow one other piece of my own advice:

Prototype, user-test, and iterate.

At Apple, we have million-dollar user-testing labs with one-way mirrors, personed by serious, white-coated psychologists with more initials after their names than can normally be found in a can of alphabet soup. We have some of the finest graphic artists in the industry.

Somehow, using all these resources and talents seemed like cheating. After all, anyone could turn out a decent song if they had Paul McCartney ghostwriting for them. I wanted to see what I could do with the kind of resources available to the average two-, three-, or ten-person shop.

Radio buttons and checkboxes have done pretty well for themselves in the last eight or so years since they first appeared on the Lisa, and I figured that this new object needed to work at least as well.

In fact, this object needed to work better, because it was not going to be appearing in any tutorial: People were going to have to be able to figure out how to use it on their own the first time out.

The behavior of the new object was a hybrid of the behavior of radio buttons and checkboxes, so my approach was to create an appearance that would reflect that marriage by combining visual elements of both.

The elements I chose were the of the checkbox and the of the radio button. They resulted in these early prototypes:

British English

British English

British English

I put together a paper test based on the following dialog box, changing only the appearance of the middle three objects between tests:

Spell Checker Preferences

Options:

- Ignore words in all caps**
- Italicize foreign words**
- Prompt for slang words**

Dictionaries to be applied:

- British English**
- American English**
- New York English**

Look for common OCR errors?

- Yes**
- No**

In addition to the new cluster of one-or-more objects in the center, please note the third question, "Look for common OCR errors?"

I have had a note glued to my monitor for months to discuss OCR (optical character recognition) and spelling checkers with y'all, because most spelling checkers, in their "suggestion" mode, fail to consider errors caused during

computer recognition—for example, errors such as the letter “m” being interpreted as “ni.”

I added this question to my dialog box so I would be reminded, as I write this, to tell all you spelling-checker guys and gals out there to talk to the OCR guys and gals when you are developing your algorithms.

I left the question in throughout every prototype (since I knew that if I took it out, I was going to have to stare at the note on my monitor for another six months) and always got the same question from the test subjects: “What’s OCR?” I was violating one of Tognazzini’s laws:




Guideline: Never present a power-user option in such a way that normal users must learn all about it in order to know that they don’t need to use it.

Back to the subject at hand: I tried the three variations of the new object on three different people in nearby cubicles, using the ten-step user-testing procedure outlined in the December ’89 and January ’90 articles on user testing in Apple Direct and my follow-up in March ’90. I first asked the users to describe the dialog box. This was to see if they’d noticed the new objects.


Then I set up a game in which they could “click” on any of the three one-or-more buttons and I would tell them what would happen “in real life” on the screen. The object of the game was to see whether they could figure out the rules governing the behavior of the new buttons.




It took the two area associates (Applespeak for secretaries) a little more than three minutes combined to figure out the rule. The third initial subject, an engineer, took just over 12 minutes to figure out the rule, but he ended up with a really neat truth table spread across his white board.





After the “game,” I showed the test subjects all three objects and asked their opinion.

They all agreed that the  was too busy. The  object showed no advantage over the . It looked less like a radio button, an association they found important.

I learned several things from this first round:

- People were having trouble recognizing the new object as a control rather than a decoration. They ascribed this to its unusual size.
- The  was working best.
- It was working best because it most resembled a radio button.
- Paper prototypes of behavioral models have limited effectiveness.


I reduced the size of the object from  to  to , at which point three additional test subjects (I never used the same person more than once for the test) clearly recognized it as a control but failed to notice that it wasn’t a radio button!

Going back to the elements differentiating a checkbox from a radio button, I focused on the straight-linedness of a checkbox, producing the following designs: , , , and .

The first I rejected as just too weird. The second fell to exclamations that it looked like a diamond with a cockroach inside. The third I rejected out of hand, because the unselected form of it would be identical to an un-checked checkbox, thereby violating the guideline against making new objects look identical to old objects. The fourth object seemed to have possibilities: It looked

buglike too (kind of like a water bug), but people didn't seem to have the visceral reaction to this particular bug that others had had to the cockroach.

I'm sure it must seem as though I had spent at least 20 or 30 hours on the project so far. Actually, I was at just under two hours. At home that weekend, I cobbled up a HyperCard version of the prototype so that I could run some decent behavioral tests.

Back at work, I ran three new people through the use of the new  object, asking them, as before, to describe the screen to find out how long it took them to recognize a new object and then asking them to figure out the rules governing the new object by playing with it.

They experienced the following problems:

- They thought that it looked like an icky bug.
- They thought that the X through a highlighted object meant that it was inactive (canceled).
- They thought that the rule was that you had to use British English. Or American English. Or New York English.


The “Superstitious” Rule

Here's what gave rise to the third problem: Users would, say, turn off everything except British English. After discovering that they could not then also turn off British English, they would make up a rule that British English was sacrosanct and could not be turned off.

Then, for the next five minutes, no matter how much I encouraged them to turn on and off buttons, they would never, ever again attempt to turn off British English.

At this point, I began to think of this design quest as merely a useful subject for a column on knowing when to abandon design quests. I saw no way out of this dilemma: Once a button had gotten “locked up” and people had made up that superstitious rule, I was sunk.

I tried making the button gray out when only one item was selected, so they would know not to push it:

 British English


I tried it on one test subject. She reported her belief that the gray meant that the selection was not available and that the British dictionary would therefore not be usable.


She also reported that my graphic looked like a dead bug. Normally, I don't put much stock in a single test subject's responses, but this outcome was pretty unequivocal. I abandoned gray.

I further abstracted the appearance, succeeding in removing the buglike appearance or making it look like a badly injured bug, depending on your point of view:

 British English

I called it the train-tunnel button and tested it in conjunction with an even further abstraction:

 British English

In tests on an additional five people, all were able to figure out the rule without getting locked up. When shown the other design, the majority favored the .

Was I ready to move ahead with validation testing (discussed below)? No, but until I came to my senses, I thought I was. After all, I had developed an appearance that people found pleasing enough and I had successfully tested it on five people. What more could anyone want?

Gee, maybe an object that actually worked.

Rationalization and Denial


I was in denial about the superstitious-rule problem: It was still there and not getting any better. So why had everyone passed my last test?

I learned early on in my prototypes always to start people out with at least two buttons on so they wouldn't get locked up on their first attempt. And now, when people did get locked up, I kept after them until they eventually tried clicking on the immutable button. "Gee, are you sure you've tried every possible combination?"

I rationalized that this activity was OK because I was not telling them directly to click on it. I was only making their lives miserable if they didn't.


This is just one of the reasons why we normally have our white-coated experts do our testing: They don't give a damn whether my design works or not, so they don't tend to bias the results. (Actually, I think that a few of them actually want my designs to fail but that they have been unwilling to admit to it up to now.)

But most developers are fresh out of white coats. I believe that you can do your own initial testing if you are able to overcome your own biases and if you can find a group of dear friends who will support you....

On a WHIM. Every week at Apple, we have a gathering of the human-interface people from around the company at a meeting called WHIM, the Working Human Interface Meeting (as opposed to the usual corporate Talking But Doing Nothing Meetings). I brought this marvelously successful new object to one of the meetings, in the hopes of roping one of the graphic designers into either endorsing or fixing up my semifinal  object. After all, my design was not bad but I do know my limitations.

They laughed, they jeered, they derided. (They are not very nice people, sometimes.) Where had I learned to draw? What was that funny line underneath the black dot?

I explained to them that the funny line was the last vestige of the X in the checkbox. They responded, "Yeah, right! I would have figured that out right away." I showed them earlier drawings, and when they spotted the cockroach and the water bug, they went wild.

When the clamor died down, John Sullivan, one of the key designers on the System 7 project, said, “If you want a cross between a checkbox and a radio button, why not make a ?” (It’s amazing the way these designers can actually speak in drawings.)

I offered a scholarly reply: “I thought of that early on, but I rejected it, because in its off state, it would look just like an turned-off checkbox. And everybody knows you should ‘invent new objects, with new appearances, for new behav...”


The whole room answered in unison: “In its off state, it is a turned-off checkbox!”

Turned-off checkbox: Clicking turns on option without affecting any other option.

Turned-off one-or-more: Clicking turns on option without affecting any other option.


The two objects did seem kind of similar, now that I thought about it. Somehow, I had just assumed that they were different.

Assumption



It had really laid me low this time: One of the earliest drawings I had made was that square plus my little insect legs: . It appeared on my scratch page of experiments along with all the other objects I’d made, and people had been drawn magnetically to it from the very beginning.

I’d kept explaining to them about how two different objects, etc., etc., etc. I’d never once revisited my original decision to reject it, even in the light of its overwhelming popularity.

I told the WHIM meeting I’d had some problems with superstitious rules but that I was going to solve them by telling developers not to actually use the thing. They gave me a series of lectures on the inherent problems of writing a column warning people not to use an object and including exacting detail on how to use it.

I installed the  and tested again. Now that I was no longer biasing the tests, people got stuck again.

Back to the WHIM meeting. A final look at the design, and then I could write up this article and tell you really, really not to use the result.

Now that they saw the  object implemented, they recognized that it was used as a radio button in another leading brand of graphic user interface. So much for the . I then broke the news that the superstitious-rule problem was worse than had I thought. I was given a new series of lectures.

As the din died down and the spitballs ceased flying, I became aware of my friend Frank Ludolph repeatedly pressing his thumb down on the table in front of him as though he were attempting to assassinate some offensive insect.

The Mercury Button. “You remember,” he said, “when we were kids, playing with a drop of mercury (before we all knew it was poisonous)? Remember how it

used to squish out of the way when you tried to press down on it? Why don't you make your buttons do that?"

We were in the presence of genius. Such a simple solution, and no one else had tumbled to it. I certainly had been walled off from ever finding it: I'd figured out two weeks earlier that the superstitious-rule problem was unsolvable.

Based on that assumption, I'd stopped trying. On an even deeper level, I knew that radio buttons don't move. A natural law of some sort.

Rationalization, assumption, and denial. I'd suffered from all of them during this project. I had also, fortunately, escaped their worst effects by doing extensive user testing and peer review. Time into the project so far? Seven hours.

I installed the mercury behavior into the prototype, following the simple rules suggested by the WHIM group: Pressing the only button turned on causes it to turn off and the button below to turn on, unless there is no button below, in which case the button above turns on.

Just Wouldn't be Natural. It seemed wooden and unnatural to chase the up from the bottom option, only to have it turn around and go back down when the next button was pressed. I made the middle button conditional: If the has been chased from below, turn the button above on; otherwise, turn on the button below. This felt much better.

I decided to push the metaphor to extremes. (I was brainstorming, and that's what you're supposed to do.) Mercury has a tendency to pop out in an unexpected direction when you press down on it, so I made the middle button pop randomly up or down.

It stunk. I tried making it even more like mercury so that if you pressed toward the top, it would squish down. If you pressed toward the bottom, it would squish upward. It was very disconcerting and still seemed kind of random. So I returned permanently to the simpler behavior, as specified here:

- If a currently turned-off one-or-more is pressed, turn it on.
- If more than one one-or-more is turned on and a turned-on one-or-more is pressed, turn it off.
- If a single one-or-more is turned on and that one-or-more is pressed, turn that one-or-more off and turn on an adjacent one-or-more, according to the following rules:
 - If the current option is other than the bottommost (or rightmost, for horizontal layouts) and the go-up flag discussed below is not set, turn on the next one-or-more down.
 - If the current option is the bottom most (or rightmost), turn on the next one-or-more up (or to the left) and set the go-up flag so that movement will now occur upward (or leftward).
- Turn off the go-up flag when
 - The top is reached.
 - A button is selected that is not currently turned on.
 - The user clicks somewhere outside the cluster of one-or-mores.

I tried the object out on my son, Joshua, age 10, and soon as he saw the ■ move, he announced the one-or-more rule. No hesitation, no wondering. Instantaneous. I figured we had a winner.

Now the only problem remaining was picking a visual appearance. For that, I submitted the prototype to Elizabeth Moller, a human-interface designer with a formal graphic-design background, who gave me back several designs, of which the WHIM group liked this one the best (note the two-pixel white border between the outer and inner diamonds):

Dictionaries to be applied:

- ◆ **British English**
- ◇ **American English**
- ◇ **New York English**

Validation Testing

You don't race around the building looking for warm bodies when you are testing for validation. Racing is OK as long as you are using a reasonably representative sample of people and are finding problems to fix, but running out of problems is not a sign that you're finished, just a sign that racing is not uncovering anything else. Running out of problems merely means that you are now ready to do some serious validation testing.

The kind of informal testing I'd done so far can be used to find problems but not to show that there aren't any. Validation testing (which is just like the rest of the testing except that you use a representative population) also finds problems, but—because it makes use of a thought-out, carefully chosen, representative sample—it can also validate a successful design.

Why the difference? Because even the first 14 people you happen to run across when you're grabbing people-down-the-hall may not come close to representing the breadth of your real target audience.

For validation testing, I chose

- A nine-year-old with limited Macintosh experience.
- A nine-year-old with no Macintosh experience.
- A woman who uses a Macintosh perhaps two hours per week.
- A woman who uses a Macintosh two hours per day.
- A male self-described power user.

This sample reflects my target population: people who would most likely have problems with the new object. This includes kids, new users, and casual users. My choice of two adult females and one adult male was random. What I was after was the casual/constant/ power-user axis.

Each of these people was able to ascertain the rules for one-or-more buttons within 15 seconds without any problems.

The object was finished.

A Solution in Search of a Problem?

“If it ain’t broke, don’t fix it”: That one-or-more buttons work does not make them a candidate for inclusion in every application from here on out. It is not particularly clear to many people here at Apple that you folks even need such an object. After all, checkboxes and radio buttons have held up pretty well. Nonetheless, my experience has been that once I have developed or heard about some new fundamental object, I find that it begins to solve a lot of problems I previously didn’t even know I had.

If you find after some time that you are using the object, let me know. If enough of you do so, we might even add it to the Guidelines.

In the meantime, I hope that you have received some insight by watching me stumble and bumble my way through this experience. (A day in the life...) And pass the liniment. It may have taken only 15 hours of design, prototyping, and testing, but it felt like 100. ◇

Send Tog (AppleLink:TOG) your questions, comments, suggestions, tirades, and editorials on the state of Apple’s human interface. Include your name (aliases acceptable), city, and state—plus company name if you wish. He will print some and pas the others on to Apple Engineering. You will be heard. He does not have the time to respond personally, but rest assured that your words will not go unnoticed.

GetNextEvent

The “ Π ” 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, write to Apple Direct Event Calendar, 20525 Mariani Avenue, Mail Stop 75-2B, Cupertino, CA 95014. For further information check the Events folder on AppleLink (path: 3rd Party Connection:Events).

November 5 through 6
The Mac Show, Philadelphia, PA
Contact: Libby Barland
(215) 540-9111
Fax: (215) 628-0882

November 11 through 14
· Autofact, Chicago, Il
Contact: SME
(313) 271-1500

November 14 through 17
CyberArts International, Pasadena, CA
Contact: Bob Gelman
(408) 446-1105

November 17 through 20
SCAMC-Symposium on Computer Apps. in Medical Care, Washington, D.C.
Contact: Gail Mutnik
(301) 657-1291

November 19 through 21
PC Expo, Chicago, Il
Contact: H.A. Bruno, Inc.
(201) 569-8542

November 26 through 29
MacWorld Expo, Austria
Contact: Patrick Dexyl
33-1-39-734042

November 27 through 29
European Technology & Issues Conference
Zermatt, Switzerland
Contact: Sandy Butler
(415) 978-3233
Fax: (415) 442-0766

December 3 through 6
· Cause '91, Anaheim, CA

Contact: Debbie Smith
(303) 449-4430

Getting Product Reviews

Target each publication individually

by John Pearce, MediaMap

Although the product review is only one of many kinds of product press coverage, it is probably the single most important element of a product publicity program.

From the standpoint of getting the right kind of information to the right people—that is, the chief buying influencers for your products—the impact of the product review can't be equaled.

Because of the function, content, and readership of reviews, their impact dwarfs that of news stories, flattering profiles of your company executives, and corporate image pieces (although those types of stories certainly have their place).

Reviews hit the bull's-eye of the market for a specific product, the real buyers: the small group of individuals that evaluate your product and either recommend it to others in their organizations—or recommend your competition.

When you look at it that way, you see that all coverage is not created equal and that product reviews can have an impact that's completely disproportionate to the number of column inches of ink you get, or even the number of readers you reach.

For example, at our company one person makes the product-purchase decisions for all our requirements. He doesn't read news stories; he thinks they're fluff, and they're not what he's looking for. He may glance at trend stories, but he does his own thinking about what products will meet our needs. And he completely ignores the personality profiles and such, because he doesn't care if someone in the industry has a colorful persona or not.

What he does care about is whether a product will meet our needs. And most often he gets that information from product reviews. He's only one reader, but he buys for 20 people. He's your bull's-eye.

WHERE ARE THE OPPORTUNITIES?

There are literally hundreds of review opportunities for computer-related products, not only in our own industry's publications but also in other media. In the U.S., roughly 70 publications dominate the computer trade press, and many of them run reviews. But although many developers limit their thinking to the trade press, there are scores of opportunities beyond the obvious ones.

One often-neglected media category is the national press, which includes major magazines and newspapers, syndicated columns, radio and TV shows, and wire services. These can be a gold mine.

One example is "Ask Dr. John," a column that is picked up by hundreds of papers of every kind and size across the U.S. Those newspapers reach literally millions of readers, many of whom are potential customers for home-use products. Because the column isn't in the computer trade-press mainstream it's easy to overlook, but it's well worth the time and effort to pursue.

Another personal favorite is the Associated Press syndicated column “CompuBug.” CompuBug runs product reviews that reach more than 400 newspapers throughout the United States.

Recently, one of our software company clients was reviewed in CompuBug; the review was picked up by hundreds of those papers—and the company was deluged with thousands of inquiries. Now, that’s cost-effective marketing!

Computer-industry newsletters. These provide even more opportunities for reviews, especially for more-technical or leading-edge products. Newsletters usually have a very small circulation compared with that of consumer publications, but they can make a huge impact. Many are expensive to subscribe to, so the subscribers—who include most of the computer industry’s opinion and market leaders—are very likely to pay attention to the newsletter’s review opinions.

Roundup Stories: The Ultimate in Reviews. The roundup story is probably the most important kind of product review, because it is a comparison of multiple products in a given category—and usually points out the best. If you come out on top, you’ll really make an impact on potential buyers.

There is no more powerful way to work with the press to help position your product. But a word of caution: As powerful as roundup stories are, if your product gets savaged or mispositioned, it can be a severe blow. As we’ll see below, avoiding such damage is ordinarily within your control or at least your influence.

Sometimes roundup stories are scheduled and listed on editorial calendars. So be sure to get the editorial calendars for all your key publications. Find out who will write the story by calling the managing editor (or referring to a resource such as MediaMap).

Once you’ve made the right contact and ascertained the direction and theme of the story, it’s time to help the editor understand how your product fits into his mission.

Comparison tables honestly contrasting and positioning your product versus others in your category can be a good way to get empirical and even subjective information across to a roundup-story editor. In any case, take personal responsibility for ensuring that the editor truly understands what your product is and is not.

Most often, however, roundup stories aren’t formally scheduled on the editorial calendar. Instead, editors look for hot market segments to cover without being constrained by a calendar planned months in advance. That’s a golden opportunity for you to pitch a roundup-story idea and is especially welcome when your company is not making any “news.”

However, you must proactively find and act on the opportunity. This is particularly true in the national press—especially the end-user press—because it isn’t hooked into the moment-by-moment news of our trade.

The key is to find a way to group your product with others and give the reviewer the appropriate information. In doing so, you help the journalist by providing a good market niche to report on and you help your company by pitching a good story that positions your product accurately.

HOW TO ATTRACT REVIEWERS' ATTENTION

At MediaMap we define public relations (including getting product reviews, of course) as “helping the press do its job.” To do that, you must understand that job and make your strategy dovetail exactly with it.

One of the biggest mistakes developers make is assuming that getting a publication to review a product works the same as getting an editor to read and run a press release—not true at all.

The product review process varies widely from publication to publication, and each may periodically change its policies and procedures. The key to getting good product reviews is understanding the differences in the product review process and editorial mission between publications.

To succeed, you must understand (and keep up with) the process for each target publication and make sure your efforts complement the publication's review process. Otherwise you'll be very frustrated. You'll send boxes of products into a black hole and wonder why you're not getting reviewed. Most of the time, it's because your approach and methods are on a different wavelength than those of the publication.

For example, if you don't do your homework for a particular magazine, you may not discover that in a given case the “product-review editor” is not an active product reviewer but is instead a front-end person or administrator, someone who receives calls from developers and says, “Yes, thank you very much, please send us your product.”

The actual reviewer may be sitting on the beach in Malibu with a portable. It's also useful to know such things as whether a publication uses a testing laboratory. Knowing that and understanding the lab's evaluation process is the key to getting a review in such publications.

For example, *Infoworld's* review process is unique. *Infoworld* doesn't use a process in which you send a package to somebody who opens it, looks at the product, and decides what the review will say. Instead, the executive editor of reviews coordinates the activities of five review teams, a testing lab, and a reader survey operation.

Each team is responsible for specific kinds of technologies, and within each team, members have designated responsibilities. The testing center works with the review teams and other editors to determine the review goals. The results of the review are combined with results from a survey of 1,000 readers to determine users' hot buttons. Needless to say, simply having a slick pitch on the phone isn't going to dovetail with the entire process.

Learn the Format, Content, and What to Send. You also should be familiar with the format and review content of a publication so that you can better position your product when you send it for review. Use that information to select the hot buttons you'll press when dealing with that publication.

Your interaction shouldn't be “Gee, won't you review our product?” It should be something more along the lines of “I hope you'll consider our product for review. Since you serve readers in the hog-farming industry, we've put together a little package of information relevant to your readers' interests, including how we stack up against manual hog-accounting techniques...” and so forth.

We recently did a major survey of the media for a client. Many editors told us that they particularly appreciate receiving a well-thought-out package of

supporting materials with review copies of products. Such materials position each product against the other products in the same niche and personalize the product features and benefits for the specific publication and audience.

You can also include backgrounders about the technology or your company. This is similar to the press-kit concept, but it's designed specifically for a product reviewer.

CREATING THE PROGRAM

Getting reviews takes planning, foresight, and some degree of commitment. So rather than approaching them helter-skelter and opportunistically, I suggest a campaign approach. Putting a working system in place and getting the first review is the most arduous part of the process (especially for smaller companies whose resources are stretched) and the most resource-intensive.

As is true with most PR efforts, the most needed resource is usually manpower. But pay that price, and you're in a great position: The incremental effort to get the next review is much smaller, and when you get the machine going and your fifth, tenth, and twentieth reviews come in, you'll realize the full power and incredible cost-effectiveness of a good product-review campaign.

Start by making someone directly responsible for the campaign. That person will plan and systematically execute the campaign and do the needed record-keeping and contact work. Then determine how many opportunities you will pursue and in what priority and order.

With more than 200 review columns to choose from, you must be selective; base your decision on who your target market is, whom the publication reaches, and the relative impact of each publication. In most markets, it is reasonable to choose 20 to 50 publications, but of course it depends on your goals and resources and how many publications reach your market.

After selecting your targets, you must research each one's review process. As I said before, each process is different—and that difference is crucial to your success. Depending on your resources and time, you may opt to do this research yourself or you can refer to research services.

Every 13 weeks, we issue a new report that describes the review process of hundreds of publications. If you opt to do it yourself, you can contact the publications and find out what their review process is by talking to the managing editor, reviews editor, or even editor-in-chief. This will take time, but you need this knowledge before you shape your strategy.

Information Necessities. You need to create a record-keeping system that tracks every review target on your list. Your system can be as simple as a form for each publication or as complex as a full-blown relational database integrated with your other PR activities, depending on your needs and resources. In any case, here's some of the information you need to record:

- For this publication, who is the audience, what is the editorial mission, and what kind of editorial posture does it generally take?
- Who are the product review contacts? What are their names, addresses, phone, and fax numbers? What is each person's role? Who are the primary versus the supporting players?
- Your form should also include space for notes about each conversation you have with editorial contact people so you'll know when you've spoken to whom

and what they said. A callback date tickler function is also very desirable for tracking your commitments to editors and for reminding you to take needed steps.

- What kind of system does the reviewer use? What computer, what operating system, what version, and what media format does the editor need? (Reviewers are not terribly inclined to rave about products they cannot use!)
- What's the lead time for reviews in this publication?
- Does the publication honor nondisclosure agreements? If so, under what conditions? What are the implications for any product-announcement schedules you may have?

With this information in hand, you are almost ready to go. Using this information and your product positioning, tailor your message to each publication's special mission and audience. Then you'll be ready to contact the editors.

STEPS TO SUCCESS

There are several steps to initiating the review campaign, all of which should probably be check boxes or blanks on your record-keeping form for each target publication.

You must make the initial contact with the editor, determine whom to ship the product to, follow up to make sure the material was received and is in good working order, and find out if the reviewer needs additional information and support. If appropriate, send reviewers guidelines or support materials, and test various approaches. Throughout the process, don't forget to keep notes of your conversations and any reactions you get from reviewers.

In addition, be sure the person responsible for your campaign has good in-house support. If the person isn't capable of handling all the responsibilities, identify and brief all the appropriate individuals, especially a technical support expert, a marketing/positioning expert, and an administrator who knows or can ascertain the status of your interaction with each target publication. These in-house players must be reliably available when needed by a reviewer.

Your campaign should also include a systematic evaluation program, a plan for making use of the feedback you get from reviews and reviewers. If you listen, the benefits are many: First, you may head off an unfavorable review before it ever sees the light of print. Second, you may get realistic feedback that will help you tailor your approach with the next reviewer, or revise your support materials or messages.

Third, you can regularly report incoming reactions back to marketing and product development so they can make adjustments in their thinking as well. Putting reviewer feedback to good use punctuates the whole process.

Timing Review Campaigns. For a new product, reviews are probably the most important press coverage you could possibly get, and you should plan your review strategy months before the actual announcement. For each target publication, devise a plan that takes into account lead times as well as nondisclosure and embargo policies.

Editors will be candid about their embargo policies (that is, agreements not to release information until a given date), and you should make the agreement explicit if you expect them to hold the review for your announcement date.

Although many will refuse to sign formal agreements, they almost always honor their commitments; an explicit verbal agreement that you follow up with a letter stating your understanding of the agreement is usually adequate.

Editors who break such agreements are quite uncommon, but misunderstandings can occur if communication is not clear.

Remember also that many publications, especially newsweeklies, explicitly do not agree to embargoes; without an explicit agreement, there's no such thing as being off the record.

The best approach is to handle the timing issue by conducting a preannouncement product tour—anywhere from a few weeks to a few months before the announcement date—featuring nondisclosure presentations to key reviewers, news editors, and analysts (again, make sure that agreements are clear). Such face-to-face interaction helps you build credibility, even in advance of your having any news to announce.

When not to solicit product reviews. It's not reasonable to expect a full review about a marginal product change. And if you cry wolf by launching an all-out product review campaign around such an upgrade, you run the risk of blowing your credibility for future efforts.

A good alternative for a less substantial upgrade or release is simply to announce it with a press release and “phone-athon” to news editors. Limit your pitch to the two or three key upgrade features that have the greatest market impact. If the upgrade is too minor to make the news pages, you can still get substantial coverage because most publications have a column for product upgrades and minor product announcements.

DEALING WITH “UNFAIR” REVIEWS

Often, developers become frustrated or irritated because they think they've been unfairly reviewed. In spite of how it occasionally may seem to an enraged product manager, editors are rarely out to kill products; they're in the business of trying to describe them accurately.

In truth, when a developer feels wronged by a review, three of every four times the responsibility rests with the developer rather than the reviewer. Your responsibility is to make sure the reviewer can't misunderstand the product.

A good example is a very high-end application—a power-user product—that gets panned by a reviewer because it wasn't easy to use. In such a case, the reviewer probably didn't understand the product's positioning. The developer must take the responsibility—and invest the effort it requires—to help the editor understand the product.

But if you feel that you've done all the right things and the review is still negative, squelch the natural desire to give the reviewer a piece of your mind or to take retaliatory action such as canceling your advertising in that publication.

Don't forget that you're in it with the press for the long haul, and act accordingly. If you feel that a reviewer has incompetently reviewed your product, by all means take action, but do so rationally.

Deal with your problem through the appropriate channels; convey your concerns to the people in charge (such as the editor-in-chief) in a reasonable and objective way.

Be very specific and concrete about how you feel your product was misrepresented. Also, many publications provide some kind of feedback forum, such as a reviews-response or letters-to-the-editor column. Take advantage of it. And realize that in the longer run your calm approach will work: If editors are incompetent, they won't be around long and you'll still have a good relationship with the rest of the editorial staff. If they are competent and have simply made a mistake, they're quite likely to bend over backward to avoid a similar error in the future.

When the Reviewer is Right. Another thought: View a less-than-glowing review as a unique opportunity to acquire useful market feedback. Usually the editors who review products do know what they're doing, and sometimes a product is criticized because it actually has a deficiency. If you consistently get hammered in a certain area, take it as a signal. One of your toughest jobs is telling your executives that perhaps the reviewer is right and that your company should act on the feedback.

Here's a story about a bad review that has a happy ending: A year or so ago, a company got a particularly bad review.

Instead of getting huffy, the company examined every criticism, went back into the labs, fixed the problems, and one year later got an editor's choice award from the publication that had panned it before.

Reviews are the driving force behind the development of the market's consensus opinion about your product. If you properly manage the process of getting reviews, the quality and quantity you get will increase dramatically—and hopefully so will your sales, company standing and image, and competitive position.◇

John Pearce is a founding partner and CEO of MediaMap, a Bedford, Massachusetts firm that provides a research service for agency and in-house PR in the computer industry.

Serving High-End Users

Cooperation is the key

by Peter Warren, Quark, Inc.

High-end corollary #1: High-end customers are very vocal. If you give them the opportunity, they'll tell you what they want and how they want it.

High-end corollary #2: Compete through technological excellence. The word will get around.

Quark aims all of its developmental effort toward serving the high-end publishing community: the magazine, newspaper, book, and advertising industries. It is gratifying to know, and interesting to note, that by aiming high, we also serve a variety of other people who use QuarkXPress for creating newsletters, proposals, brochures, and—yes—even word processing. The message here is clear. Some people who don't have complex design and production requirements still want and need functionality beyond what is available from word-processing and entry-level publishing programs. The same concept could well hold true for users in your target market segment.

Customers just starting out may not need all of your high-level features immediately, but they may need them in the future and will buy your product now. Some may be attracted to the level of sophistication your product offers.

Or they may simply want to use the tools "that the big kids use." "Why not spend a little extra and get the best?" this customer will say. If your product's interface is clean and easy to use and understand, there's no reason why lower-end users shouldn't also be able to take advantage of what you can offer.

THE COMPETITIVE EDGE

Reaching all levels of users, though, requires satisfying the needs of the high end with a technologically sound product. Because of the demands high-end customers place on your products, competing through technological excellence is imperative. You must understand how your users think and work. Therefore, becoming intimately involved in what your market does and how it does it will give you a competitive edge.

It's a given that high-end users will press Macintosh technology to the limit. Their work is creative in nature, and they will seek new and different ways to handle a problem. They are generally enamored by the productivity that computer technology offers and will adapt quickly to take advantage of a full range of hardware and software. This is especially true in the Macintosh environment, where people traditionally use more applications and peripheral devices than do users of other platforms.

High-end users are a finicky, demanding, vocal lot. Those in high-end markets scream, demand, argue, insist, cajole, and scold. They always want more features than you can reasonably deliver. And well they should! Professionals in high-end markets are not simply using your software, they are intimate with it.

Think about it. Chances are they are spending more time with your product than they are with just about anything else. Their livelihood often depends on your product's promised performance. And if your product line delivers the solutions they need, it's likely your users will become part of your family, so to speak. As family members, users are proud, loyal, and dedicated. They want you to succeed. And if you listen to them and provide them with viable options, they will help assure your success.

A TALE FOR OUR TIMES

Here's a modern-day parable: An art director of a major monthly magazine recently told me that after his staff had made the transition from traditional design and layout systems to electronic publishing on the Macintosh, an interesting thing happened: The staff members realized that they could now do things never before possible and could instantly see the results of their labors.

Their design options increased exponentially, and they were delighted with the way their creativity was enhanced with this newfound technology. As time passed and his staff became "fluent" in QuarkXPress, he noticed that they began to take advantage of every product feature. They began to create increasingly complex page designs.

The moral of this story: If you give professionals a productivity tool, they will push it as far as they can.

However, when a product is pushed to its limit, something else begins to happen: Users begin to pose questions such as "If it can do this, why can't it do that?" and "Boy, this is stupid. Can't they do a better job of implementing this feature?"

This is good. Sure, they can jam up your phone lines, but they will help you with product design and give you a good sense of what they think and how they use the product. You should use high-end users' tendency to be vocal to your advantage.

You won't be able to satisfy their every request, though. They want absolutely everything!

As your products grow, you must carefully select which features to add. Otherwise your product becomes overloaded with features, which distracts from performance and users' productivity. When a product is loaded with features that appeal only to highly vertical markets, it will be ignored by the masses.

KEEP THE DOORS OPEN!

The question becomes: How can you modify your application to meet users' increasingly sophisticated demands and desires while keeping the program manageable and the unit price down? There are several approaches. As your market matures and grows, it's a given that you'll need to add features. However, this entails significant trade-offs. When you add features, you do so at the expense of increasing memory requirements, program complexity, and cost. Unfortunately, the only thing you risk decreasing is program speed.

Another way to extend functionality without adding giant hunks of code to the base application is to make your product extensible, by allowing add-on software modules. By doing so, you can keep your program from becoming

unwieldy. More importantly, you can satisfy the specific needs of a small group of power users without sacrificing your larger market opportunity.

For example, as we prepared to release the first version of QuarkXPress, we realized that no matter how robust we made the program, we could never meet all the diverse needs of the publishing community with a single product. The design, layout, and production needs of design firms, newspapers, corporations, magazines, directory and catalog publishers, and advertising agencies all are very different from one another. So instead, we set out to develop a sophisticated, highly precise product to meet a broad range of common needs for the high-end professional publishing community.

To do this, we developed XTensions technology. XTensions are software modules, created by either Quark or other developers, that add new features and capabilities to QuarkXPress.

With help from the Quark research-and-development staff, these developers have built front- and back-end applications designed to ease the burden of common publishing tasks. Most XTensions add functions that are not available through Quark-XPress. Common XTension applications include pagination systems, classified-ad production and management, database publishing, and links to scanners.

Because XTensions “nest” within QuarkXPress itself, they actually become part of the program. When you launch QuarkXPress, XTensions are integrated into the program, providing users with solutions to problems unique to their environment. Although there are currently 43 commercially available XTensions, some publishing problems are so specific that publishers themselves will become XTension developers; they build their own XTensions to meet their unique needs.

This strategy has been an enormous help in our gaining high-end market share. We believe so strongly in XTensions technology that we are creating an XTension program for our editorial-workgroup product, Quark CopyDesk, currently under development. And we already have developers committed to building XTensions!

Driving Sales Indirectly. We see no direct revenue from XTensions created by independent developers. But if an XTension meets a customer’s need, it will help drive the sales of Quark-XPress. An extension to your product may be the critical make-it-or-break-it factor for your users. If your product by itself can solve part of the problem but your product extended can bring it all home, then you have met the customers’ needs. Opening your system eliminates the need for customers to reluctantly adapt to how your product works.

(And here’s my opportunity for a commercial plug: If any of you have a good idea for an XTension, let us know. For more information, contact XTensions, Quark, Inc., 300 South Jackson, Denver, CO 80209, fax (303) 377-6327, AppleLink: D1590. Those of you outside the U.S. can contact Q.S.S Kilbarry House, Dublin Hill, Cork, Ireland, fax (021) 300171, AppleLink: D2351.)

If the extensibility needs for your product are modest, you will probably find that a specially crafted set of Apple events will do the trick for you. In cases where the communications bandwidth between your product and the extension product are relatively low and when it is not desirable for the extension to make fundamental alterations to product behavior, Apple events can handle most

problems nicely. For those other cases, another interface may be desirable. If you craft such an interface properly, you can even add support for additional Apple events via this lower-level extension capability.

[Editor's Note: Some other developers, such as Interleaf, SoftSync and Shana, do provide for "fundamental alterations to product behavior" via Apple events. For further information on programming for Apple events, see "Scripting Your Success," in the September '91 issue of *Apple Direct*, and "System 7 Success Stories," in the August '91 issue.]

CONNECTIVITY THROUGH COOPERATION

Connectivity with other software and other platforms is another way to leverage your product's functionality to better serve high-end users, who tend to use complementary software and hardware. Complying with industry standards works in your favor by enabling users to integrate a variety of products.

End users will ultimately determine which products are technologically the best in terms of making life easier and more productive. Consequently, rather than forcing users to use numerous proprietary systems, it makes sense for developers to work together to create an environment that's conducive to building integrated solutions.

Competition is good, but cooperation is too. An integrated-solutions approach is appealing because it doesn't lock your customers into a single (expensive) product and it lets those technology-sensitive users take advantage of the range of cutting-edge technologies.

This is where standards come into play. The most recent Publishing Interchange Language (PIL) specification, for instance, allows software and hardware companies to exchange page-description files between programs and across platforms.

A standard such as this offers the publishing community greater freedom than ever before. It's good for the market, and what's good for the market is generally good for developers.

Moreover, extensible applications combined with standards allow developers to add product features that make their software more responsive and productive.

These products have a more broad-based customer appeal, without being lumbering giants that are hard to use and even harder to understand.

KEEP YOUR EAR TO THE GROUND

To build a product suite that will sell to the high-end market, you need to stay on top of what the market needs. Because the feature set that will satisfy this market is a great deal more complex than one for lower-end markets, keeping in touch should be an ongoing process. There are several ways to gather market information.

To develop a relationship with your installed base, encourage the formation of independent local and national user groups. (They don't have to be enormous to be beneficial.) These forums will provide opportunities for your users to get together to solve common problems. Experienced users are often willing to help those just getting started with your software.

This kind of support is invaluable to all parties. User groups are a great source of feedback from your most important customers. You benefit by getting a sense of what users are doing with your software, what they are having problems with, and what they would like to see in upgrades or new products. In return, you can help sponsor some of the meetings, provide lists of users for membership mailings, and help sustain the user group (and thus your customers and potential customers) in a variety of other ways.

Electronic bulletin boards are an excellent and inexpensive way to solicit comments and suggestions from your market. High-end users are more likely to subscribe to such services than are mass-market users, making this an excellent forum. At Quark we have a team of people who regularly monitor CompuServe, AppleLink, America OnLine, and Connect. They answer questions about our products and pose questions to the market. For instance, we post questions asking what new features users would like to see in QuarkXPress — whether a proposed new file format is important to them, if a new XTension makes sense, and so on — to get a feel for what they are thinking. We find that it is well worth the time and effort, because it gives us instantaneous feedback.

Bulletin boards are also an excellent way to monitor what your competitors are doing and how the market is responding to their products. Mass negative reaction to something your competition is doing can often present an opportunity for a new product.

Disk-Based Feedback. One of the best ways to keep track of your market is simply to ask (remember, this is a vocal group). On the registration card, ask customers some basic questions about who they are and what they do. You can also inquire about where they get information, to help you fine-tune your marketing-communications program.

Here's Quark's approach: To install QuarkXPress, you begin with a disk that asks basic name and address information. We then ask a few simple questions about what your industry is, the kind of work you do, and which publications you read on a regular basis. This kind of information can be instrumental in determining future product direction. We offer 90 days of free technical support to users who send in the disk. At our end, the information is downloaded to a database.

Tim Gill, Quark's founder and senior vice president of research and development, reads all the comments that come in through either bulletin boards, registration disks, or the mail—as do several members of the R&D department, product managers, and marketing managers. There is no comment too insignificant!

Qualitative and quantitative surveys are also good ways to gain information about new-product development. You don't have to spend huge amounts of money to get useful information. Just keep it simple, and you won't have to hire a big-ticket market-research firm. The July and August issues of *Apple Direct* contain very informative articles on qualitative and quantitative market-research methods you can use yourself.

Did I Say That? If you listen carefully, developing and selling your product to high-end markets is easy. I can't believe I just said that! It's never easy, but it

can be a lot less painful if you pay attention to the market. Users will tell you what you they want and how they want it configured. But you do have make a concerted effort to solicit opinions.

My advice: Remember the two corollaries, and put them to use. This crowd is vocal; keep listening. Use what you learn. And if you continue to offer upgrades, provide ways to customize your product to meet their needs, and develop new products — all based on technological excellence — they will remain loyal friends.

And friends tell friends. ◇

Peter Warren is the marketing-communications manager at Quark, Inc., Denver, Colorado.

Pacific Market Forum 1992

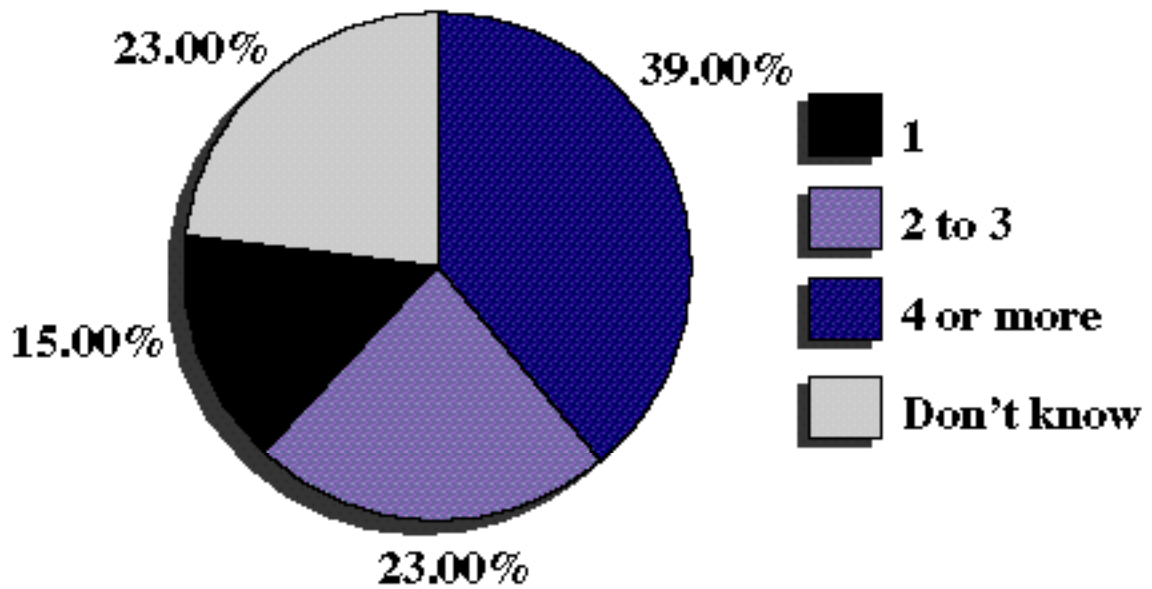
If you're one of the 400 developers who attended last January's Pacific Market Forum, you had a chance to hear about Apple's strategy in its Pacific Rim markets and meet with developers and distributors from the region.

The Pacific Market Forum 1992 will provide more of the same, with special sessions for developers who are already marketing their products abroad. Overall, the conference will concentrate on market-development issues and market-entry strategies.

The Forum will take place January 16 and 17, right after Macworld Expo, in the Fairmont Hotel in San Francisco. The registration cost is \$450 for all preregistered attendees. If you haven't received an invitation or would like more information, please send a fax to Pacific Market Forum 1992, at (408) 974-6622, or write to Pacific Market Forum 1992, Apple Computer, Inc., 20525 Mariani Ave., Mail-Stop 29-H, Cupertino, CA 95014.

How Many Have Four or More Megs?

As you'd expect, Macintosh customers are using systems with more memory than ever before. The chart above shows the percentage of Macintosh users who report having a specific number of megs in their systems; the data comes from Apple's 1991 Macintosh user study. The percentage of users reporting four or more megabytes has grown from 22 percent in Apple's Macintosh 1989 user survey to today's 39%.



Source: Apple Computer, Inc. 1991

It Shipped

Through the "It Shipped" program you can announce new and revised third-party products in *Apple Direct*. It Shipped listings are also made available on the 3rd Party Connection AppleLink bulletin board. You can obtain an It Shipped application by downloading it from the AppleLink network: (AppleLink path: Developer Support:Developer Services:Developer Programs: It Shipped Program OR ThirdParties: 3rd Party Connection: It Shipped!)

Once you complete the application, send it to: Engineering Support, Apple Computer, Inc., 20525 Mariani Ave., M/S 42-ES, Cupertino, CA 95014 Attn: It Shipped Program. Or send it via AppleLink to IT.SHIPPED

Optionally, you may wish to send us a copy of your product to be placed in the Engineering Support Library where it may be checked out by Apple's testing groups for compatibility testing or by Research and Development employees for evaluation. If you would like your product(s) to be included in the Engineering Support Library, send your product(s) to the address above.

Publisher	Product (Version)
Athena Computer Corporation	LYNX™ (1.0)
Cambridge Computer Corp.	mxConnect (4.0)
Casa Blanca Works, Inc.	Drive7 (1.5.1)
Casady & Greene, Inc.	Aqua Blooper Piper (1.0)
Casady & Greene, Inc.	Crash Barrier (1.0)
Casady & Greene, Inc.	Fluent Laser Fonts Complete Cyrillic Library (4.05)
Casady & Greene, Inc.	Fluent Laser Fonts Library (4.05)
Casady & Greene, Inc.	Fluent Laser Fonts TrueType Starter Set (4.01)
Casady & Greene, Inc.	Glider (4.0)
Casady & Greene, Inc.	QuickDEX II (2.0)
Dow Jones & Co., Inc.	Market Analyzer PLUS (2.0)
DYNAWARE	DynaPerspective 2.3 (2.3)
Golden Triangle	DiskMaker (1.0)
Golden Triangle	TwinIt (1.0)
ICOM Simulations, Inc.	TMON Professional (System 7 compatible) (3.01)
Ides, Inc.	Selector (5.01)
Koala Acquisitions Inc.	MacVision Color Video Digitizer (1.1)
Koala Acquisitions Inc.	MacVision Image Processing Software (4.0)
Logicat Inc.	Passmaster (1.1)
Neon Software	RouterCheck™ (1.0)
Rebo Research	Mix By ReStore™ (1.0)
Rebo Research	Mixer FKey (1.0)
Rebo Research	ReStore™ (2.0)

Rebo Research
Shield Microcomputing
SoftWriters, Inc.
Systat, Inc.
Techware, Inc.

Warner New Media

ReStore™/Pro (1.1)
Hydraulics Test (1.0)
VersionTerritory (2.0)
Network Mac SYSTAT (5.1)
IRAQ STACK, Hypermedia Software on
the Gulf War (1.0)
Brahms "A German Requiem" (N/A)

APPLE DIRECT



Apple Direct is Apple's monthly developer newspaper, covering business and technical issues for decision-makers and development companies. It is published by Apple Computer, Inc's. Developer Support Systems and Communications (DSSC) group.

Publication team:

EDITOR-IN-CHIEF:

Lisa Raleigh (AppleLink:RALEIGH.L)

TECHNICAL WRITER/EDITOR:

Gregg Williams (GREGGW)

BUSINESS AND DEPARTMENTS EDITOR:

Dee Kiamy (KIAMY)

PRODUCTION MANAGER & DESIGNER:

Hartley G. Lesser (H.LESSER)

CONTRIBUTORS:

Jessa Vartanian, Suzanne Dills, Sharon Flowers, Monica Meffert

MANAGER, DSSC:

David A. Krathwohl

FILM:

Aptos Post, Aptos, CA

PREPRESS:

*Prepress Assembly,
San Francisco, CA*

PRINTER:

Wolfer Press, Los Angeles, CA

APDA, Apple, the Apple logo, AppleLink, AppleShare, AppleTalk, A/UX, LaserWriter, LocalTalk, Macintosh, MacApp, MacTCP, MPW, SANE, and SADE are trademarks of Apple Computer, Inc., registered in the U.S. and other countries. FinePrint, HyperScan, Macintosh Quadra, MacX25, OneScanner, PhotoGrade, PowerBook, QuickTime, ResEdit, Sound Manager, SourceBug, SuperDrive, and TrueType are trademarks of Apple Computer, Inc. Classic is a registered trademark licensed to Apple Computer, Inc.

PostScript is a trademark of Adobe Systems, Inc., registered in the U.S. and other countries. NuBus is a trademark of Texas Instruments.