

Inside this issue

TECHNOLOGY & INDUSTRY

WHAT'S IN STORE FOR 1992 FROM APPLE? Kirk Loevner, director of the Apple Developer Group, gives a glimpse of what's ahead and reviews 1991. (See What's in Store for '92 folder)

A/UX IS EVOLVING to PowerOpen, beginning with the release of A/UX 3.0 (see News folder)

A SNEAK PREVIEW of the "New & Improved Inside Macintosh" is on this month's Developer CD.

There's also a new document viewer. Apple wants your feedback on both. (see News folder)

WANTED: DEVELOPER FEEDBACK on the new mailing. We've received many comments from developers and would like to receive even more. o PowerOpen, beginning with the release of A/UX 3.0 (see News folder)

MACWORLD EXPO will be the venue for Apple Developer Central, a set of activities sponsored by the Apple Developer Group, which includes a DTS lab, training courses, a software tools exhibit, and more...(see News folder)

MACHACK '92 is still several months away, but it's not too early to start thinking about it if you're interested in presenting a paper there. In fact, the event's promoters have issued a call for papers. (see News folder)

THE RESEMBLANCE is strictly intentional...the folder layout on the Developer Service Bulletin Board has been revamped to more closely parallel the folder structure on the Developer CD. (see News folder)

NEED A CONTRACT programmer? The new version of the Macintosh Services Directory might be able to help. Sent to Associates and Partners in the November mailing, the guide is also available from the MZ Group. (see News folder)

BESIDES THE NEW *Inside Macintosh* chapters and their accompanying “viewer,” this month’s CD also features several new international system software versions, new hacks, printing software, and the summaries of two recent market research studies on benchmarks and notebook usability. (see News folder)

USER GROUPS, of course, are a great word-of-mouth resource for developers. To make it easier for user groups and developers to make connections with each other, a Speaker’s Bureau has been formed that helps developers find a ready audience. (see News folder)

APPLE’S ANTIPIRACY efforts have included a focus group that solicits developer input, discussions with MIS directors, and a new AppleLink forum on the topic. (see News folder)

VIRTUAL USER V. 1.1 and the *Apple Publications Style Guide* were recently released by APDA. (see News folder)

BUSINESS & MARKETING

BRODERBUND SAYS the key to compatibility testing is doing it not only the right way, but also at the right time. Only then will you get the quality-control that the market demands. (See Business/Marketing folder)

UPGRADE BLACK holes can be avoided if you take fully into account the effect that offering an upgrade will have on your entire business. One option, says Jordan Levy, is to hire an upgrade “campaign manager.” (See Business/Marketing folder)

NOTEBOOK USABILITY was the subject of a study by GVO, an independent research firm, which found the PowerBooks to be the most usable in a field of ten notebook-class computers. (See Business/Marketing folder)

BOTH THE CD and AppleLink now feature marketing folders, with information such as international market guides, *Apple Direct* marketing articles, product data sheets, and market research data. (See Business/Marketing folder)

Inside Macintosh Sneak Preview

This month's Developer CD contains two big surprises: advance previews of the next generation of *Inside Macintosh* documentation and Apple's viewer software for on-line technical documentation. (These are currently named "New & Improved Inside Macintosh" and "Inside Mac Viewer," respectively, but both will eventually get different official names.)

The files are located on the December CD, *The Silence of the ROMs*, Vol. XI, using the pathname Dev.CD Vol. XI:Technical Documentation:Inside Macintosh Preview.

The "New & Improved Inside Macintosh" folder contains approximately 1,000 pages of preliminary documentation from 14 chapters of *Inside Macintosh* in 6 files. Before reading these files, be sure to read the "IM Viewer Read Me," "About These Files" (in the folder with the chapters), and "IM Viewer User Guide" files.

Because both the documentation and the viewer are in very early form, Apple wants and needs your feedback. To provide feedback on the contents of "New & Improved Inside Macintosh," please send your comments via AppleLink to DEV.TECH.PUB, and to give us feedback on the viewer, send your comments to AppleLink address DEV.CD. Apple is planning to release additional chapters in several future 1992 Developer CDs. Addison-Wesley will publish the first volumes of "New & Improved Inside Macintosh" late in the summer of 1992.

Apple Direct will keep you informed on future details of both the documentation and the viewer as they become available.

Feedback Wanted on New Mailing

Since October, when we made substantial changes to the monthly Developer Mailing, we have received both positive and negative feedback on it from many developers. We take your feedback very seriously and would really like to know what more of you think about the monthly CD, the new format for *Apple Direct*, and the new emphasis on electronic information delivery.

We encourage you to send us your ideas, suggestions, and comments on any aspect of the Developer Mailing. Please send your input via AppleLink to DEVHOTLINE.

What's in Store for '92

And a look back on '91

by Kirk Loevner, Director, Apple Developer Group

Change has always been a constant at Apple, and 1991 was no exception. In the past year, we've seen fundamental changes in Apple's business model, a major reorganization of the company, significant new business alliances, and of course, the introduction of innovative new products—both from Apple and from third-party developers. Although change is never easy, I believe that our efforts over the past year have set the stage for Apple's as well as developers' continued and increasing success throughout the coming decade.

As we head into the new year, I think it's important to take a few moments to reflect back on some of 1991's highlights—to get a sense of how far we've come. Then I'd like to share some insights on where the Apple Developer Group (ADG) is headed in 1992. I'll tell you about our top priorities for the year, give you a glimpse of some new products and services in the ADG pipeline, and tell you about some new initiatives we'll be undertaking to enhance the economics of Macintosh software development. 1992 is going to be an exciting year for all of us.

GLANCING BACK AT 1991

Think back a year or so. What was on your mind back then? For many of you, it was a time to take a good hard look at your business and decide where it's headed and where you should invest your resources. Just like Apple. Maybe you were wondering if System 7 would ever go golden

master and if Apple's market-share strategy was really going to work; most importantly, you wanted to know where Apple was heading in the future. "QuickTime" probably didn't mean much to you, Apple's notebook computer was still only a dream, and the possibility of Apple and IBM joining forces was almost unfathomable.

It certainly has been a banner year for change. A few of the highlights follow.

Market-Share Strategy. The biggest change at Apple in the past year has been the refocusing our business strategy. Our fundamental business model changed from one that emphasized high gross margins to one that seeks profitable long-term growth through increased unit shipments and market share.

To succeed long-term, we needed to increase our market share and become a much bigger player in mainstream computing. This new strategy, combined with challenging economic conditions, meant that we had to manage our expenses more closely and be selective with our investments.

And the strategy is working! Worldwide unit shipments of Macintosh computers increased more than 60 percent in our fiscal year 1991, ending in September 1991. In an industry that experienced overall unit growth of only about 3 percent this past year, our success is indisputable.

This phenomenal unit growth has translated directly into gains in market share, which has increased from 10 to 15 percent in the U.S. in the past year, according to third-party analysts.

We're also making great strides in European and Pacific markets. Third parties report that Apple has taken the lead in terms of unit market share in both France and Sweden, pulling us up to the number 3 position overall in Europe.

In the Pacific region, we doubled unit shipments in Japan this past year, as we did in 1990. We expect this trend to continue, given the tremendous acceptance of the PowerBooks in Japan.

For developers, this means larger markets for your products. Developers consistently tell us that the most valuable thing we can do for them is to increase the unit volumes and installed base of Macintosh computers. And that remains our primary goal.

Partnering and Licensing. This year, the words *alliance* and *license* showed up in Apple press releases more frequently than in the past. The Apple/IBM alliance was certainly the most talked about, but we've also partnered with IBM and Motorola on the new RISC PowerPC architecture and with Information Builders, Inc., in the client/server-software area, providing a Macintosh solution for accessing nonrelational data from a wide variety of databases.

Last month Apple Europe signed a letter of intent with Digital Equipment whereby Digital will sell Macintosh products to its customers as part of its own solutions and Apple in turn will provide Digital interconnect products, VAX and RISC-based servers, and a range of desktop services to customers through its Apple Centers in Europe.

Through the AppleTalk Licensing Program, we've supplied networking source code to more than 90 licensees. We've also signed several licensing agreements for the Data Access Language (DAL) source code.

No business has the resources or expertise to go it alone in the 1990s. By combining our unique strengths with the complementary strengths of other companies, we'll not only serve our customers better but we'll also ensure our long-term success as a business. That's why we will continue to focus on our core competencies and partner with others when it makes sense.

Time to Market. Apple reorganized into separate hardware and software divisions last April to allow for faster time to market and formed new business groups to take advantage of emerging opportunities such as enterprise systems and consumer products. We've been very successful in decreasing time to market over the past year. In the past three months alone, we've introduced more CPU products than in any previous year in our history.

Full Product Line. As we start the new year, Apple's product line has never been stronger.

We now have entries in all segments of the market, from the low-end Macintosh Classic and LC systems, to the midrange Macintosh II family, to the '040-based Quadras, and the PowerBook notebooks. We also have a broad range of printing and imaging solutions available, including the new Apple OneScanner, and a wide variety of monitors. With the thousands of software applications available for the Macintosh, customers can now fill all their personal-computing needs by mixing and matching your software and our hardware. Whatever the need, together we can fill it.

Developer Support. Developers tell us that they want to be self-sufficient. You prefer being able to find the answers to your own questions rather than having to ask us. And you don't want to look in a lot of places to find that answer or spend a lot of time or money in the process. New developers tell us that they'd love some help in just getting started.

That's why one of our main focus areas this year was on expanding the suite of self-support resources, lowering barriers to entry in Macintosh programming, and helping guide new developers in getting started.

A few examples of our efforts in this area include the *Macintosh Services Directory*; the Dev Tech Answers library on AppleLink; self-paced training products from Developer University, including the new "Introduction to Object-Oriented Programming" course; the comprehensive E.T.O. (Essentials•Tools•Objects) CD from APDA; the Dev Info Assistant on the Developer CD Series; and APDA's Developer Resource Kit and Getting Started bundles.

We continue to take a global approach to developer support. We're doing more to ensure that our materials are more accessible and understandable in countries outside the U.S., not to mention an increased focus on international topics in *Apple Direct*. Maintaining strong working relationships with Apple developer- support organizations worldwide remains a top priority.

Map to the Future. “What’s Apple’s future direction?” is a question developers have always asked. This year we mapped out our long-term product strategy that will take us through the rest of the decade.

Following the directional statements made alongside the October 2, 1991, Apple/IBM announcement, we published the *Blueprint for the Decade* booklet, which provides an overview of Apple technology and strategies, including our plans to rapidly advance our System 7 software architecture; it also discusses where we’re headed with RISC, object-based systems, and enterprise computing. (U.S. and Canadian Associates and Partners will find *Blueprint for the Decade* in this month’s Developer Mailing. Developers outside the U.S. should contact their local office for information.)

Armed with this information, you’ll be better prepared to plan for your own future as an Apple developer, and together we’ll be able to more closely align our strategies.

ON TO 1992

Third-party products from Apple developers are what really bring the Macintosh alive for customers, and the most innovative products continue to be developed for the Macintosh. Our charter in the Apple Developer Group is “to drive the development and success of superior third-party products on Apple platforms.” To this end, we will spend more than \$60 million next year on developer tools and support. Read on for a few highlights of where we’ll be investing those resources.

System 7 and Extensions. For many, 1991 will be remembered as the year Apple shipped System 7. *Apple events*, *IAC*, *Balloon Help*, and *file sharing* are just a few of the terms that have become commonplace among the development community. Adding to the 100 or so System 7-savvy applications that were available at introduction time, developers worldwide continue to bring

System 7-savvy applications to the marketplace. And we will continue to advance System 7 throughout the 1990s.

We will be focusing our system-software efforts on integrating support for multiple media types (e.g., QuickTime) as well as delivering on the promise of a true set of collaborative system-software services, including IAC and messaging. We feel that these will be two of the biggest areas of differentiation for us, and we will pursue them vigorously. In addition, we will be improving the “global readiness” of our system software and make it easier for you to reach international markets.

We’ve also evolved the way we deliver system software. Whereas in the past, major advances in system software meant releasing entirely new versions, this year we’re moving to a strategy of modular extensions to the System 7 base system.

This means that Apple can ship new technologies right away (instead of making the world wait for the next “big bang”-type release) and that you can integrate them into your products just as quickly. QuickTime is one of the first examples of this new modular strategy, and you’ll see future expansions throughout the coming year.

OOP and Development Tools. Development tools are a major focus for the coming year.

A top priority will be to enhance our object-oriented programming (OOP) tools. One of the best examples of how OOP can simplify complex tasks and increase programmer productivity is demonstrated in how easily MacApp developers have been able to take advantage of System 7 features. MacApp 3.0, which includes extensive support for System 7, ships this winter.

In addition to enjoying the immediate benefits of adopting an application framework such as MacApp, developers who begin the migration to OOP today will also be more competitively positioned to exploit the Apple/IBM joint venture Taligent a few years down the road.

We’re also continuing to improve our core development environment, MPW. We will provide incremental linking, among other performance enhancements, during 1992. In the longer term,

we're in the process of shifting to a new compiler technology that will provide you with world-class compilers for our coming RISC architecture as well as for our 68000-family systems.

Cross-Platform Development. Most of you develop for other platforms in addition to the Macintosh. So this year, Apple is putting together a cross-platform development strategy that will allow you to more easily target other platforms while fully exploiting the advantages of the Macintosh. Our strategy addresses cross-platform issues in four areas: interoperability, system-software APIs, data standards, and development tools. Specific developer-requested pieces of Apple technology, such as QuickTime, will be part of these plans. You will see progress on all four fronts in the coming year—we'll be sharing more information with you as the strategy evolves.

Application Scripting. Building upon the Apple-events foundation of System 7 as well as the efforts of developers who have added support for Apple events and AETE (Apple Event Terminology Extension), Apple will introduce application-scripting tools in 1992. We will also continue working with third-party scripting-tool developers. For more on application scripting, see "Scripting Your Success," in the Sept. 1991 issue of *Apple Direct*.

Emerging Opportunities. Another question developers often ask us is where we see opportunities in the marketplace. A few of the major areas are: media-integration products, collaborative applications, and portable-specific computing applications.

Media integration is a great opportunity because, we believe, time-based data manipulation via QuickTime will soon become just as widespread and important as QuickDraw was in the early days of the Macintosh. And now is the right time to start building it in to your products.

Collaborative and workgroup applications will also become very important as Macintosh gains ground in enterprise-computing environments—demand will increase for applications that help people work together more effectively. Mail, messaging, and telephony are just a few of the things to explore in this arena.

Also, with the success of Apple's PowerBook notebook computers and our IAC and Open Collaboration Environment (OCE) technologies, customers are looking for solutions that let them do their computing anytime and anywhere.

These customers have unique needs, from mobile and deferred communications to vertical applications (real estate, for example) to who knows what else—the field is wide open for innovation and creativity—and we're looking forward to seeing the amazing things you create.

Business Development. The "success" portion of our charter will be a big push this year. In the past, we've focused on helping you develop great products, but oftentimes those same products have trouble attaining acceptance and success in the market. That's why this year we want to go beyond being just a development partner to being more of a business partner with you.

We want to improve the economics of software development by increasing the revenue potential of Macintosh software and also by reducing the costs of developing a Macintosh product.

We are approaching this area from two sides. First, we are looking at ways that Apple can help influence the environment you operate in. And second, we're focusing on ways we can help support you in your own business development.

A good example of one of our first efforts in this area was the marketing-consultant meetings at the 1991 Worldwide Developers Conference. We brought in independent marketing consultants and matched them up with developers for one-on-one consulting sessions, free of charge.

The sessions focused on whatever topic the developer wanted to discuss: PR, distribution, packaging, pricing, or something else entirely. And since the meetings were confidential and one on one, the developers received consultation on their specific issues and situations. Overall, developer feedback on these sessions was very positive.

Not only did developers get some hands-on advice but they also got to test-drive a consultant's services with the only investment being their time. We'll be providing more services like these in the future.

Antipiracy. One big opportunity for increasing the revenue potential of Macintosh software can be found in fighting software piracy—which is why we undertook an antipiracy initiative earlier this year—and it will continue to be a major theme throughout 1992.

Piracy dramatically affects developers' success and profitability. The Business Software Alliance (BSA) estimates that worldwide, piracy cost the industry \$10–\$12 billion in revenues in 1990. Imagine the effect on your business's bottom line were piracy dramatically reduced...

Our antipiracy campaign has several elements. First we're launching an education campaign, targeting a wide variety of audiences, including user groups, dealers, customers, and—of course—developers. We're also working closely with groups around the world, such as the Software Publishers Association (SPA) in North America, the Japanese SPA (JSPA), and the BSA, that have an interest in curtailing piracy.

Another aspect of our initiative is exploring Apple and third-party security strategies from a technology perspective. As we make progress, we'll bring you updates here in *Apple Direct*.

For the latest on our antipiracy efforts, see "Update on Apple's Antipiracy Campaign" on page 6 of this issue.

Developer Support. Creating new self-support tools and resources will continue to be a priority for us in 1992. We know you're anxiously awaiting the revised *Inside Macintosh*, and you'll find the first chapters of the revised edition on this month's Developer CD Series, with a new application for viewing them on line. We're also continuing to enhance the user interface on the Developer CD Series, making it even easier to navigate, bringing more consistency to file formats, and generally doing more to help you find exactly what you're looking for right away.

Other things in the works include a new self-paced training product from Developer University—“AppleTalk for Programmers.” Developer University will also deliver the “Macintosh Development Tools Advisor,” a terrific new resource for developers who need help selecting the right tool for the job.

To help new Macintosh programmers get up and running, we’ve just published a “Roadmap to Macintosh Programming” flier, outlining key Apple and third-party resources that both new and experienced programmers might find helpful throughout the various steps of the development process. For the more advanced programmer, Developer University has two other courses on the way: the “Advanced System 7” and “Intermediate Object-Oriented Programming and Design in C++” courses.

Staying in Touch. Developer satisfaction is a key goal for all of us in ADG. To that end, we do a lot of surveys and focus groups to stay in touch with how we’re doing in supporting you.

To all of you who have taken the time to fill out those surveys or to exercise your vocal chords over the phone or in focus groups—thanks. And although we’ve never worried about Apple developers being soft-spoken, we do hope you’ll continue our dialogue throughout 1992.

Looking back over the past year really points out how far we’ve come and what we’ve accomplished together. It has truly been a year of change—change that was tough at times but that ultimately has set the stage for an exciting future. And although change will always be a constant here at Apple, one thing you can always count on is our commitment to developers.

It is my hope that this glimpse of where ADG is headed in 1992 will make you better able to synchronize your development with ours. Together, we’re going full steam ahead. ◆

A/UX Evolving to PowerOpen

First step is A/UX 3.0

What does the Apple/IBM alliance mean for you and your customers? In the case of A/UX, Apple's enhanced implementation of UNIX, the answer is clear: A/UX has a strong future as it evolves into PowerOpen, the new open-computing environment that will be supported by Apple and IBM. Equally importantly, PowerOpen will not make your current efforts with A/UX obsolete.

A/UX 3.0 brings System 7 technology (and many other features) to Apple's integrated Macintosh/UNIX operating environment. This new release of A/UX also provides a high-performance environment for Apple's new Macintosh Quadra 700 and 900 computers. Apple will release A/UX 3.0 sometime during the first calendar quarter of 1992; the price has not been set.

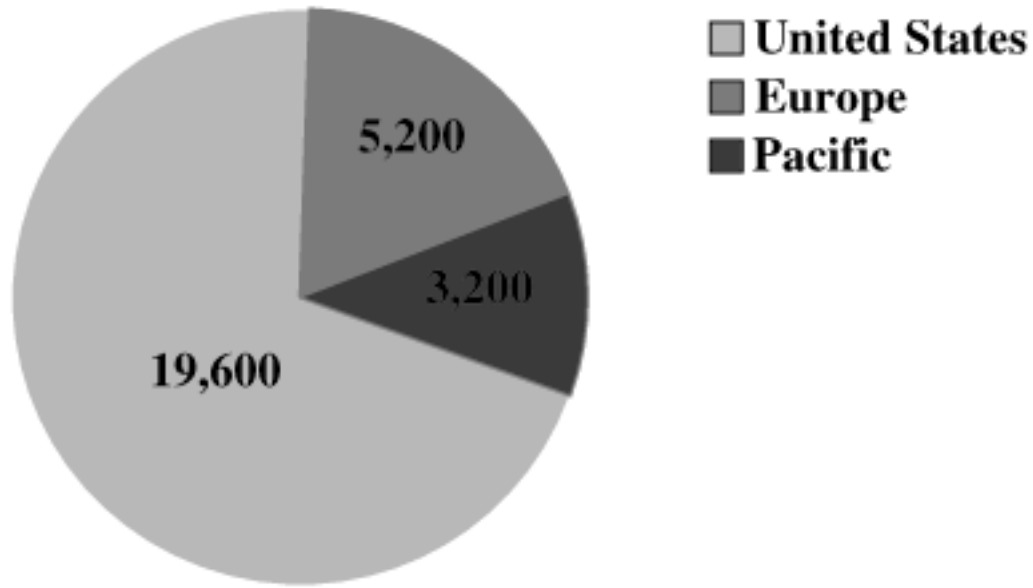
Apple wants you to know that the evolution of the A/UX operating environment includes backward compatibility: The same software that now runs on A/UX will be usable with PowerOpen. Apple and IBM will each continue to market their own versions of the UNIX operating system, and Apple will further differentiate A/UX by adding innovative and powerful Macintosh technologies to its industry-standard version of UNIX.

A/UX 3.0 is the first step in Apple's transition to the shared PowerOpen technology. Sometime in the late-1992/early-1993 time frame, Apple will deliver a later version of A/UX 3.x that will add more Macintosh system- software enhancements, new hardware support, and increased AIX compatibility.

Sometime in late 1993/early 1994, Apple will offer A/UX 4.0, its PowerOpen system. A/UX 4.0 will run on the PowerPC RISC-based hardware platform as well as on

the Motorola 68000-family hardware platforms. The same Macintosh applications and AIX applications available today will run atop the OSF/1-based PowerOpen platform.

Apple Developers Worldwide by Region



As of May 1991, there were approximately 28,000 Apple developers worldwide. This includes all types—commercial developers, in-house/custom developers, systems integrators, software houses, university/research developers, etc. The total of 19,600 developers in the U.S. reflects U.S. APDA customers, including U.S. Associates and Partners.

Source: Apple Developer Group Research, May 1991.

MacHack '92 Call for Papers

Even while winter is settling in, high-end Macintosh developers are casting their thoughts ahead to June 1992, when MacHack will bring its seventh annual conference to Ann Arbor, Michigan.

Organizers of the conference have issued a call for papers to be presented during technical sessions of MacHack '92. No more than ten papers will be presented; past topics have ranged from "Future Mac Evolution" to "Parallel Processing Paradigms."

Supported by the University of Michigan, The MacTechnics Users' Group, and Apple Computer, Inc., MacHack has become well known for its informal, dynamic, and sometimes flat-out rowdy atmosphere.

The conference is also well known for its "Proceedings" optical discs, issued for the past two years and containing many of the entries from the conference "hack" contest. The contest, sponsored by the MacHax Group, forms a focal point for MacHack and encourages developers to use their creativity and technical ability. It has annually produced a mix of useful, fun, and sometimes awe-inspiring tools and utilities.

Examples include the infamous "Net Bunny" and the humorous "Oscar" singing trash can.

Containing more than 400 megabytes of tools and utilities, including more than 100 "hacks" from the past four conferences, the annual disc has become a prized item for developers. Copies of the proceedings have spread MacHack's reputation internationally, with discs going to several countries outside the U.S.

MacHack has always relied on limiting its size to keep the conference informal and to encourage interaction among attendees. In 1991, there were 260 attendees, and the 1992 conference size will be limited to 300. As an opportunity for developers to share

work in progress, MacHack has always been on the cutting edge of Mac technology. The database-management program 4th Dimension was introduced at MacHack '87, and the conference's all-night machine room has seen some very new technology made available to Mac devotees.

MacHack '92 will be held June 17–20 at the Ann Arbor Holiday Inn, West. Discs may be ordered (\$50 includes shipping) and advance conference registrations may be made through Expotech, Inc., 1264 Bedford Road, Grosse Pointe Park, MI 48230; (313) 824-7992. AppleLink: EXPOTECH.

Abstracts of papers to be submitted for presentation at MacHack '92 should be sent in either hard copy or electronically to Waldemar Horwat, 9 Fort Washington Place, Cambridge, MA 02139, or via Internet to waldemar@hx.lcs.mit.edu. ◆

Developer Events at Macworld Expo

Whether you're a seasoned Macintosh programmer or an aspiring one, Apple Developer Central will be the place to be during the Macworld Expo in San Francisco next month.

There will be a full agenda of developer-related activities sponsored by the Apple Developer Group, including tools sessions, a lab hosted by Developer Technical Support, training courses, a software tools exhibit, APDA information, demos of new self-paced training products, and information on the Apple Events Developer Association.

You'll find Apple Developer Central in Moscone Center in Rooms 232, 236, and 238 on the East Mezzanine Level. Admission is free to all Macworld Expo attendees—a conference badge is not required. Hours will be 12–6 P.M. on Sunday, January 12; 10–6 on Monday and Tuesday, January 13 and 14; and 10–4, Wednesday, January 15.

Some of the highlights will be:

- *Tools Exhibit.* Whether you're working at a beginning, intermediate, or advanced level, the professionals you meet and the products you see at the Apple Software Development Tools Exhibit can enrich and expedite your development effort.

Representatives of leading tools publishers for the Macintosh will demonstrate their compilers and utilities, answer your technical questions, and listen to your feedback.

- *Developer University Hands-On Training.* Come try the newest offerings from Apple Developer University: multimedia, self-paced programming courses. The courses take full advantage of DU's instructional experience and provide complete learning environments, up to and including building real applications.

Three self-paced courses will be featured: "Macintosh Programming Fundamentals"; "An Introduction to Object-Oriented Programming"; and a preview of "AppleTalk for Programmers," which will be released this spring. Also featured will be the new

“Macintosh Development Tools Advisor” reference tool, an electronic guide to Macintosh development tools.

- *APDA Tools*. APDA offers convenient access to more than 300 development tools and resources for anyone interested in developing applications on Apple platforms. Come see us at the APDA information station—we’ll show you a range of tools and resources that can help get you started in Macintosh programming or keep you successfully going. Ask us about APDA’s special Macworld promotion coupons!

- *Apple Events Developer Association*. The Apple Events Developer Association (AEDA) works with the developer community to define event standards. Drop by to see how you can get started with Apple events and participate in event standardization. Bring your ideas for scenarios and events.

- *Developer Technical Support Lab*. Apple Developer Technical Support will conduct a limited number of one-on-one debugging labs at Macworld Expo. Meetings will be by reservation only—first come, first served. Reservations will be taken in the debugging room, number 232, located in Moscone Center. You do not need to be an Apple Associate or Partner to take advantage of this service.

DTS provides engineering consulting for Apple’s commercial developers on software and hardware products for Apple computers. You are welcome to bring in your source code or just come in and discuss issues.

- *Sessions*. There will also be technical sessions held on the following topics:
 - Choosing a Macintosh Programming Tool
 - Macintosh Programming Fundamentals
 - A Tool for All Seasons (learn about tools to support the complete software-development cycle)
 - MacApp 3 + System 7 = Perfect 10
 - Moving to MacApp 3.0
 - Macintosh Common Lisp in Action

- Becoming System 7 Savvy
- Symantec's THINK Languages
- System 7: The Basics and Beyond
- MPW: The Next Generation
- Introduction to Object-Oriented Design
- Introduction to C++

For further information on the sessions, look in the Developer Events folder on AppleLink (Path—Developer Support: Dev-eloper Services:Apple Information Resources:Developer Events). ◆

Speakers' Bureau Up & Running

Apple's User Group Connection and the Mac User Group News Service (MNS) recently joined forces to offer speaking opportunities across the U.S., giving developers the chance to address user-group audiences.

Since its inception in July, the Speakers' Bureau has made several good matches, according to MNS head Don Rittner. For instance, Kensington Microware's product managers called, wanting to know what user groups were meeting in Chicago while they were in town. With little effort, Rittner found them an audience.

The Speakers' Bureau can respond to such requests because Rittner maintains a database of user groups and their meeting schedules. In addition, he keeps a current file of developers, including speaker profiles, product lists and descriptions, preferred topics, and any travel restrictions. When he gets a call from a company looking for a chance to reach the top ten user groups, as Lotus recently did, he can provide the particulars.

The Bureau also offers flexibility. Developers sometimes call at the last minute, wanting a speaking engagement or demo opportunity while they're on the road. Others scout for a match a few months ahead of time—which is, of course, the preferred way to go.

Apple User Group Connection manager Craig Elliott has described the setup as a “win-win situation” and emphasizes that Apple's support of the bureau is another way to make it “as easy as possible for developers and user groups to work together.”

Prospective speakers and interested user groups can register by completing an application form. To do so, contact Don Rittner through MNS on America On line (AFL DonR), AppleLink (UG0194), or GEnie (MNS), or call (518) 374-1088. ◆

Developer CD Dec. Highlights

Each month this column will be your guide to the latest Developer Series CD, telling you what's new and notable. To quickly access everything listed below, see the "What's New on This CD" folder (located at Dev.CD Vol. XI:Start Here/Read CD License 1st:What's new on this CD?). Here you'll find aliases to every new and updated package on the CD, including all those listed below. The November CD—*Silence of the ROMs*—includes the following highlights:

Inside Macintosh preview with new viewer: Thanks to the diligent efforts of our outstanding technical-publications team, we are proud to bring you a preview of the first chapters from the "New & Improved Inside Macintosh" (working title). Also included with this package is our new viewer tool. This tool lets you open, search, and print the enclosed text documents without requiring any additional software on your part. This is prerelease software, so beware—you will find bugs. Please send us your feedback on how you like the viewer, since we are seriously considering using this tool for other materials on the CD. The chapters previewed on this disc include "Collaborative Computing," "Devices," "Imaging," "Macintosh Toolbox," "Operating System," and "Worldwide Software."

Int'l system software. This CD features new versions of international system software, including: Arabic 6.1, British 7.0.1, ChineseTalk II 1.0, Danish 7.0.1, French 7.0.1, French Canadian 7.0.1, German 7.0.1, Greek 6.0.7.1, Hebrew 6.1, International 7.0.1, Italian 7.0.1, Kanji 6.0.7.1, Norwegian 7.0.1, and Spanish 7.0.1.

Market research. *Silence of the ROMs* includes two new marketing studies to assist you in evaluating Apple's new machines. The first, the Ingram benchmark study from October '91, provides a summary of the key findings of a performance study conducted by Ingram Laboratories. The study tested a variety of personal computers running applications available for both the Macintosh and Windows environments and found that the Macintosh Quadras were the fastest of all. The second study, the GVO Notebook Usability Report, provides the findings of a notebook-computer usability study performed by GVO, a leading industrial-design company. The PowerBook 170 and 100 came out on top in this survey (see page 10 of this issue of *Apple Direct* for a summary).

Hacks and applications. This CD brings you several assorted new tools. FinderHeaps is a debugging tool that allows you to track the Finder's heap sizes. It includes the Application Heap, the System Heap, the Code Heap, and the Object Heap. HyperQ. Plastics and HyperDesign. Plastics are tools to aid in the selection of plastic materials for mechanical design.

Printing software: Several of our printing tools, including LaserWriter, StyleWriter, and Personal LaserWriter software, have been updated. There are also several new printing-related snippets among the December snippets. Be sure to take a look at DTS Groupies, a snippet that groups pictures and takes them apart, using PicComments and a QuickDraw bottleneck procedure. ◆

DSBB Gets a New Folder Layout

Parallels CD folder structure

If you're a regular AppleLink user, you may have noticed that Apple is restructuring the organization of information on the Developer Services Bulletin Board (DSBB), located in AppleLink's Developer Support area. Our goal is to have the information structure on the DSBB match that of the Developer CD, allowing you to navigate with ease regardless of whether you're on the CD or on AppleLink.

We plan to have the new DSBB layout completely done by January. The information on the Developer CD, then, will be a subset of the information available on the DSBB. For example, you'll find "Headlines for Developers" on the DSBB but not on the CD. In addition, the DSBB will house historical information from past CDs, so if something drops off the CD because of space constraints, you can always find it on the DSBB.

The other difference between the DSBB and the Developer CD is that we have the capability of restricting information by audience here on the DSBB, as we did when we published information on the alpha and beta versions of System 7 and restricted access to developers only.

The new top-level structure of the DSBB will look like this:

- New Information
- Headlines for Developers
- Apple Information Resources
- Developer Essentials
- Development Platforms
- Marketing Information
- Periodicals

- System Software
- Technical Publications
- Tools & Apps

The new structure for the DSBB will be implemented folder by folder—with approximately one new folder every other week. While we're restructuring, the existing folder structure will remain in place. Whenever you open an "old" folder, you'll see a pointer to the information's new location.

One final reminder: We'd like to encourage you to use the BB Pathfinder, available in the AppleLink Information Area, to search for documents whenever you're unable to find them on your own. The BB Pathfinder is a great tool to help you locate information. If you have any questions about this restructuring, please send a Link to TONI.T. ◆

New Macintosh Services Directory Available

Let's say the bulk of your coding is done and it looks as if there will be a sizable European market for your product. Where can you find a reliable service to run some independent tests before it's out the door? How can you get your hands on a decent software translator in a hurry? And who on earth can train your tech-support staff before orders pour in?

For answers to these and many other development-quandary questions, take a look at the Fall 1991 edition of the *Macintosh Services Directory*. The new volume updates and adds to the store of information found in the first edition, published last spring.

Entries now number more than 1,300—about 40 percent over last time.

There are several new features that make the new directory more useful. In particular, there's a fuller set of categories, now including database programming, system integration, multimedia, networking, and user-interface design, among others. All categories now also feature more-descriptive subheads, which is especially helpful as specialization increases.

Application development, for example, is now broken into areas such as business, education, publishing, and entertainment. Another new element: small ads that allow developers and consultants to better differentiate their services. And the geographic reach is broad, with entries representing most U.S. states and a dozen countries.

The Fall '91 directory was sent to all certified Apple Associates and Partners in the November Developer Mailing. You can also order a printed copy, a copy on disk, or bulk copies by calling the MZ Group at (800) 927-1100, ext. 400.

The single-copy price is \$14.95 (\$17.95 outside the U.S.); the disk costs \$395.

If you're interested in appearing in the spring '92 edition of the Directory, call Helen Arrick at MZ Group ([415] 543-8290) or contact her on Applelink at MZGROUP. ◆

Update on Apple's Antipiracy Campaign

Developer input sought

Apple Computer is continuing its efforts to combat software piracy by organizing an antipiracy brainstorming session with developers, creating an antipiracy discussion area on AppleLink, and releasing French and Spanish versions of the Software Publishers Association (SPA) antipiracy video.

Software piracy, the illegal copying of software, cost developers an estimated \$2.4 billion in the U.S. last year, according to the SPA. Apple has spearheaded a campaign to mitigate this industrywide problem in the coming year. The campaign continues this month with an Apple internal awareness campaign that began on November 26, 1991.

Developers Share Ideas. At a brainstorming session last month, 18 developer representatives from a variety of markets were invited to share their ideas on fighting piracy. More than a hundred ideas—suggestions ranging from awareness campaigns to technology-based solutions—were generated in the session. In the next few months, Apple will be evaluating and sharing these ideas with developers. Here's a sneak preview of some ideas ranked as the most promising by the group:

- *Education Programs.* Session participants unanimously felt that antipiracy education programs for end users would be highly beneficial. Users need to be educated about respecting the intellectual copyrights of software developers. The consensus was that Apple should investigate ways to help provide education and raise awareness in these areas.

- *Software Registration Incentives.* Providing materials and services that can be received only by registered customers is an important way to discourage piracy. Essential benefits should be good support, manuals, and upgrade notices. Other suggested incentives included inexpensive

upgrades, auxiliary software given away to registered customers only, and informative user newsletters or magazines.

- *Registration Amnesty Program.* Software companies can build their base of legitimate users by offering “pirate” users inexpensive “no questions asked” upgrades. This tactic has worked extremely well for Symmetry Software, creator of the ACTA-7 Outliner program. The company placed a small ad in *MacWEEK*, offering illegal ACTA users inexpensive \$50 upgrades if they sent in a sample screen shot.

The company increased its customer base by an impressive amount and, surprisingly, received many apologetic letters from ex-pirates. As a result, Symmetry will be rolling out a similar program when it releases its KanjiTalk version this year.

- *Network Application Servers.* A user-counting server application would allow a network administrator to control the number of concurrent users using a given application. Many developers felt that it was important for Apple to set standards for this kind of technology by recommending a specific application or building the technology into system software.

Discussions with MIS Directors. At Apple’s MacIS Conference last month, 225 MIS representatives from major corporations such as Arco, TRW, Kodak, and Union Carbide discussed the issue of software piracy.

Members voiced strong support for Apple’s initiative in combating piracy and emphasized the need for the SPA and Apple to influence the Macintosh development community to create a standard, easy-to-understand software-license agreement.

Many of the attendees’ companies are currently conducting aggressive campaigns against piracy and are looking forward to the release of the Macintosh version of the SPA’s software-auditing application in early 1992.

Ordering the SPA Antipiracy Video. Apple has just completed French and Spanish translations of the SPA’s antipiracy video, “It’s Just Not Worth the Risk.” This 12-minute video is

designed to educate companies on copyright law and the legal use of software. It dramatizes the penalties, lawsuits, and negative publicity that a fictional company suffers from after it's caught in the act of software piracy. To order copies, contact the Software Publishers Association, 1101 Connecticut Avenue, N.W., Suite 901, Washington, DC 20036; (202) 452-1600; fax (202) 223-8756; or in Paris La Défense at 33-1-4692-2703.

Apple's Internal Campaign. On November 26, Apple's COO, Michael Spindler, kicked off an internal awareness campaign to remind employees of Apple's policies on legal software use and to make sure that all software is properly licensed. Developers are encouraged to initiate similar campaigns in their own companies.

Antipiracy Discussion Forum. Apple would like to hear about your antipiracy success stories and ideas on fighting piracy, through the new Antipiracy Discussion on AppleLink (path—Developer Support:DeveloperTalk:Antipiracy Discussion). This is where we'll keep you posted on the results of the brainstorming session and general news on software piracy.

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:Apple Information Resources: Developer Program Information:It Shipped! Program). Or contact Todd Luchette at (408) 974-1241 (voice) or (408) 974-3770 (fax). Once you've completed 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 them to the address above.

Publisher Product (Version)

U.S.A.

Aladdin Systems, Inc.	Shortcut (1.5.2)
AT&T Graphics Software Labs	StudioMaster (1.0)
Avocat Systems, Ltd.	Law Office Manager (2.3)
Concentrix Technology, Inc.	Connections (2.0)
Evergreen Technologies, Inc.	MedVision (1.0)
Franklin Estimating Systems	Estimator for Macintosh (1991-2.0)
GBA, Inc.	SalesFORCE—The Proposal Generator (1.1)

Imaja	Chronos (1.0)
Language Systems	FORTTRAN Tools for AppMaker (1.1)
Mac Software Registry, The	The Mac Software Registry Database (1.0)
MacLaboratory, Inc.	MacLaboratory Psychology (2.0)
	MacLaboratory Psychology: Psychological Disorders (1.0)
	MultiDrive (1.4)
	SpeechWare (2.0)
Magna	Empower I (4.0.7)
	Empower II (4.0.7)
Masters Publishing	The Programmer's Extender GT (1.0)
Microseeds Publishing, Inc.	HAM (1.0)
Mission Mountain Software	O'Tools (3.0)
Novell	NetWare for Macintosh (3.01)
ON Technology, Inc.	On Location (2.0)
PC Globe, Inc.	MacGlobe (1.0)
PEMD Education Group	PEMD Discovery U.S. Economic Data CD-ROM Disc (1.0)
Radius, Inc.	PrecisionColor/24X
	PrecisionColor/8X (upgradable) PrecisionColor Display/20
Second Wave, Inc.	Expanse NB8 (1991)
ShirtPocket Software	SpeedyCD (1.1)
Stone House Systems, Inc.	T•View (1.0)

QuickTime Correction

There was a slight timing error in the “QuickTime Goes Final” article on page 1 of last month’s *Apple Direct*. We stated, “Apple has completed the final QuickTime 1.0 release. The INIT is available on AppleLink (path—Developer Support: Developer Services: Developer Technical Support: QuickTime). And next month, a QuickTime Development Kit will become available through APDA.”

Apple actually expects to make the QuickTime INIT available on AppleLink sometime in December. As soon as it’s available, we’ll post a notice in the Headlines for Developers folder, alerting you to it.

In addition, developers will receive the QuickTime 1.0 CD in December, and APDA will announce the availability of the QuickTime Development Kit sometime in December. We apologize for any inconvenience the error may have caused you.

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:Apple Information Resources: APDA—Tools for Developers).

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 also tells 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.

Apple Publications Style Guide, Fall 1991 Edition

\$30.00

A7G0030/D

Virtual User v. 1.1

\$50.00

M0987LL/C

APDA TOP TEN SELLERS

1. E.T.O: Essentials•Tools•Objects
2. Data Access Language (DAL) v.1.3
3. *Inside Macintosh*, Vol. 1-6, + X-Ref
4. Macintosh Programming Fundamentals
(self-paced training)
5. MacTCP 1.1 Developer's Kit
6. Macintosh Common Lisp v.2.0b1
7. MPW C and Object Pascal Bundles
8. MPW C++ v.3.1
9. MacX25
10. ResEdit v.2.1.1

*as of 11/19/91

To place an APDA order from inside the U.S., contact APDA at (800) 282-2732. APDA's number in Canada is (800) 637-0029. And for those who'd like to call the U.S. APDA office from abroad, the number is (408) 562-3910. 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, see the "International APDA Programs" page in the most recent *APDA Tools Catalog*.

On High-Altitude Computing

By Tog Tognazzini

I am writing this column at 50,000 feet on an airplane. (Well, strictly speaking, *in* an airplane.

Things haven't gotten that austere at Apple.) I am using a PowerBook 170, which I managed to get on loan in the guise of doing human-interface research. So far on this trip, I've researched the problems of playing PGA Golf at high altitude. It doesn't seem to have improved my game, but I do seem to be able to loft the ball a little better.

After two weeks of living with a portable, I have discovered that my style of using the computer has changed—because of the environments in which I have been attempting to use the computer and because of limitations of the trackball. These have led to my depending much more on the keyboard than I have previously.

(Trackballs are not bad devices, but they just don't do as well as mice. On the other hand, mice are not too practical at the back end of the plane here in cattle class, as you are liable to elbow the football player squeezed into the seat next to you in your efforts to drop a document into the trash. Which could give a new meaning to the term *catastrophic data loss*.)

Here are a few preliminary guidelines that will make your applications portable-savvy and justify my further playing of golf on airplanes. You'll notice that some of these observations can apply to nonportable applications as well:

Guideline: Expect users to make more errors.

Portables are used outside of well-designed work spaces. They are held on people's laps on airplanes, trains, rapid-transit systems, and even buses. People will be making mistakes, and our applications need to be particularly sensitive to what kinds of errors can occur, heading off the errors when possible and providing simple, straightforward recovery when they occur anyway.

Guideline: Expect users to make more errors when selecting from menus.

Make sure that users who select the wrong item will become aware that they have done so. For example, do not make the menu items Open and Delete Document appear adjacent to each other and then offer virtually identical dialog boxes that require the same user actions.

Guideline: Test applications on the portable by using subjects who are advanced in years or otherwise less physically able than the average user.

Any physical limitations of the user are magnified during trackball use—reduced coordination, hand tremor, and the like make precision movement extremely difficult. We are a young industry, and most of you are far younger than many of your users. The fact that you can wheel around the interface with no problem is in no way predictive of whether others will also be able to run your applications with the portable's trackball. Pay careful attention to actions that cannot be carried out successfully by test subjects, and either reduce the need for precision or offer efficient, effective keyboard alternatives.

Guideline: Applications must be forgiving of fingers' losing contact with the trackball button during drag operations.

The user's fingers can lose contact with the trackball button because of a peculiarity of trackballs: With a mouse, the fingers holding the mouse and the finger perched over the mouse button move in concert as the mouse moves over the physical desktop. With the trackball, the palm or fingers moving the ball move relative to the fixed position of the finger on the trackball button. Even with the extended size of the Apple trackball buttons, it is easy for the fingers to momentarily lift away from the switch during the effort to manipulate the trackball during a drag operation.

Guideline: Applications must be forgiving of incorrect initial mouse movement, as the trackball is difficult to control during "takeoff."

With the trackball, the initial direction is difficult or impossible for many people to control. This problem becomes particularly acute in some graphics programs.

Guideline: Graphics programs that support the Shift key for constraining mouse direction must be fully forgiving of users' starting off in the wrong direction.

This guideline has always seemed to me a matter of common sense, yet many paint programs continue to set the constraint direction with the first movement to an adjacent pixel. These applications are impossible for many portable owners to use. (I tried to use one—just released two months ago—and was completely unable to predict the direction a constrained line would take.)

Where practical, algorithms should not lock in a constraining direction until the moment the mouse is released. For example, if the user has drawn a three-inch-long horizontal line and, while still holding down the mouse button, moves much above 22.5 degrees, you should jump the line up to a 45-degree angle.

Guideline: If memory is available, applications should reside in memory, instead of paging in, to save on power (and users' time) by not spinning up the hard disk all the time.

Where it is impractical to have an entire application in memory, applications should be very carefully segmented to reduce hard-disk access to a minimum.

One strategy for reducing the apparent time to hard-disk spin-up is to keep in memory enough of each logical segment to keep the user occupied while the balance of the segment is brought in. For example, in a spreadsheet application that can create a chart, hit the user instantly with the dialog box requesting information about what kind of chart is desired, allowing you the time to bring in the charting segment while the user is trying to figure out the difference between a bar and column chart.

Code for enabling users to pull down menus and so on should always be available: It becomes quite frustrating when it takes almost five seconds just to pull down the File menu. Keeping these small menu resources in memory, preloaded and nonpurgeable, is a good trade-off, although each developer needs to analyze which menus make most sense to keep loaded and ready to go.

A person can live with a notebook portable, traveling with it everywhere. It opens up uses for the computer that were either not practical before or were at least less likely to be needed before.

Here are a few I'd like to have right about now.

The electronic daybook. Yes, yes, I know—we have electronic calendars popping up everywhere. But most of them are oriented toward desktop use, where the calendar is an adjunct to the rest of the user's information life. These calendars have the advantage of being networked to all my fellow worker's calendars, but they do me no good when I'm up here in cattle class. There I must either depend on printed-out copies of my schedule or carry a copy of my calendar with me on my notebook.

The problem is, even if I carry it with me, it will not yet replace the functionality I get from my paper calendaring system. My paper calendaring system is divided into four primary sections: calendar, running to-do list, address book, and project sheets. The calendar functions are well duplicated by current Macintosh applications. The to-do list is also covered by individual applications, as are the address book and even the project sheets, which I can maintain in my word processor. But what I am looking for is an integration of all these functions into a single application. Otherwise, I am just as well off carrying around my paper system, which never has low batteries and has a rather smaller theft potential.

Integration would offer me several important benefits. As it is now, most new information I enter into my running to-do list: I copy each voice-mail message into the list and then respond to it. If I set up a meeting with a person I've called, I copy the name onto the calendar. If the person turns out to be someone I will want to contact in the future, I copy the name and number into the phone list. If they become an eventual contact for a project I'm working on, I copy the name and number onto a project sheet. Every time I make one of these copies, I risk a transcription error—not too important when I'm writing down the name for a meeting but very important when I'm transferring a phone number into the address book.

In a fully integrated notebook application, all these various forms I now use would be views of the same information. I would never have to copy down anything. (If caller-ID catches on, even the original phone number will not have to be written down, given a port from voice mail into the Macintosh.)

Slide presentation. I have landed in Texas. The conference I'm attending has recessed while our hosts reconfigure our meeting room for lunch. Most everyone is off on a nature hike, attempting to sidestep the sidewinders and the fire ants. I have a portable Macintosh, so I can get some work done. Is this progress?

Slide-creation programs abound on the Macintosh, and all contain a presentation mode, whose chief job it is to turn off the menu bar. Most, if not all, have carefully duplicated every limitation of the mechanical slide projectors that preceded them, with a slavish adherence not seen since the days of the Glass Teletype, when early video terminals duplicated every aspect of the mechanical machines they were replacing.

Mechanical slide projectors are linear. Slides are stacked in a predetermined order, and every slide is presented in that order. The presenter can back up to a previous slide and skip over subsequent slides by briefly flashing them on the screen. Other than that, whatever order was chosen during creation is the order in which they will be shown during presentation. This is often a most unfortunate model.

All of us at one time or another have sat through presentations that were originally assembled for a different audience with different needs. In the worst instances, the presenter greets each and every slide with a disclaimer about why this slide is not particularly valid in this instance. Is this lack of organization the result of poor planning on the part of the presenter? Sure, but it happens often, because of the extreme burden the correct planning requires.

As a professional speaker, I have historically avoided slides, just because of the limitations of linear presentation. I want a tool that enables me to show the slides I want in the order I want, and I want to make those decisions during the course of my presentation.

Essentially, I want to keep my slides in outline form, so that if I want to dwell on a specific subject for a specific audience, I can draw forth supporting material. On other subjects, where I detect a lack of audience interest, I can skip the deeper material, still knowing that it is there in case relevant questions arise at the end of the talk.

Data updating and transfer. The portable needs to be integrated not only into the user's life but also into the user's information space. I now spend four days per year laboriously going through every file on my home and office systems, updating each system to the other. With the portable, this task will grow even worse. That users should even have to perform such a task is unfortunate: What I am doing in pawing through every folder in a half-gigabyte of hard disks is a purely mechanical process that the computer would be quite qualified to do on its own.

Ideally, I would like to get the portable near my home or office system and have them start flinging information back and forth. Failing that, I would like one single LocalTalk cable with which I could tie them together, along with a single utility I could launch that would cross-update my systems, prompting me only where human judgment is required.

Individual applications such as calendars need to support data updating too. When the portable is connected to my work computer (by phone or in person), the two calendars need to update each other.

It's your turn. Let's get these products on the market, pronto. And let me know what other products you think are needed for the portables, or just what you think of portables in general. My Link address remains: TOG. ◆

If you have information about a show 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— Developer Support:Developer Services:AppleInformation Resources:Developer Events).

January 12 through 15

△ **Macworld Expo**,
San Francisco, CA
Contact: Mitch Hall & Assoc.
(617) 361-8000

January 17 through 19

NAMM-National Association of Music Merchants
Anaheim, CA
Contact: NAMM
(619) 438-8001

January 22 through 24

△ **Uniforum**, San Francisco, CA
Contact: PEMCO
(800) 323-5155

January 23 through 24

△ **Developer Kaigi**
(Apple Japan Developer Conference)
Yokohama, Japan
Contact: Eiko Kohara
(03) 5562-6230
Fax: (03) 5562-6060

January 29 through 31

△ **Page '92 (DTP Trade Show)**
Tokyo, Japan
Contact: Ichiro Onishi
(03) 5562-6250
Fax: (03) 5562-6060

February 12 through 14

GTC Southwest
Austin, TX
Contact: Yubi Wahlquist
(916) 452-4902

February 14 through 18

△ **NASSP—Nat'l Association of Secondary School Principals**
San Francisco, CA
Contact: Karen Kordes
(703) 860-0200

The Upgrade Black Hole

by Jordan Levy, Upgrade Corporation of America

Why some upgrade campaigns get stalled

First, the good news: Upgrades usually can generate a flood of money from a company's installed base. In 1990 the software industry generated more than five percent of its revenues through selling upgrades—\$1 billion, according to the Software Publishers Association.

Projections indicate that by 1992 this number will rise to \$2 billion, making the upgrade business virtually an industry unto itself. (These numbers include “predatory” or “sidegrade” sales—that is, sales made to a competitor's installed base.)

Now the for the bad news. The opportunities are alluring, but there's another side of the coin: The same hoped-for influx of orders can crash your phone system, bring chaos to the shipping department, and antagonize and alienate thousands of your once-loyal customers.

THE CHALLENGE OF UPGRADE SELLING

Selling upgrades entails a special kind of marketing that is at the same time easier and more difficult than selling the previous version of a product. You've got the advantage of selling to someone who has used the product, so you have a ready-made audience for your pitch. However, it will be difficult to sell the upgrade to any customers who didn't like the product.

Even if their experience with the product was satisfying, users may still decide that they don't need the upgrade. In such a case, you have a real job to convince them that they need the added features; if you can show that you're truly adding value to the product, you'll get a better overall response. (I've been very interested to note that, at least in my experience, Macintosh customers generally seem to upgrade at higher rates than do DOS users.)

In addition, in many instances, the channel was involved in selling the original product; the reseller(s) took a piece of the pie. Upgrades, on the other hand, can be sold directly to customers. Therein lies the rub: Many developers who successfully market initial versions of a product find themselves in a quandary when it comes to selling upgrades.

If you don't understand the process and the resources it can consume—and if you don't have a well-planned, well-executed upgrade campaign plan based on the resources at your disposal—you risk wreaking havoc on your organization and your customers.

Successful upgrade sales depend on your knowing how to avoid several obstacles that often get such campaigns into trouble. One key decision is whether to do the campaigns yourself or to hire an outside firm to handle part or all of the job (see the "How to Hire a Campaign Manager" sidebar). Here are some potential problem areas to troubleshoot if you opt to do it yourself

ABANDONED CALLS

An upgrade offer usually results in a surge of calls that can overwhelm a company's phone system. Typically, half of the response to your campaign will come via your toll-free phone lines. If the volume of calls exceeds the system's capacity, you'll aggravate

a lot of people. Moreover, roughly 15 percent of people who get blocked or abandoned the first time never call back, and those sales are lost forever. You must also make sure that you have enough people staffing the phones. Callers generally won't wait more than 30 seconds to talk with a sales representative.

As an aside, I'd like to point out the importance of giving customers a toll-free telephone number. Making customers pay for a call (or for a postage stamp, for that matter) is an excellent sales prevention tactic; it dramatically reduces the response rate. If you don't have a toll-free number, chances are that you'll lose much more money in sales than you would ever have spent on postage and phone calls.

Having the appropriate telephone equipment, the right number of incoming lines, and the right kind and number of people staffing them can help avoid this problem. However, the real key lies in prevention—through planning. Often, when companies are planning upgrade campaigns the drive or desire to “just get the revenue flowing” is so strong that it gets in the way of managing the campaign the right way. You have to put the horse in front of the cart; you can't get the revenue if you can't handle the traffic. So you should do some creative and resourceful planning.

For instance, if you have 10,000 customers on your upgrade list (thus 10,000 upgrade offers to make), you can stage the mailings instead of sending them out all at once (and suffering the resulting phone jam). Instead, mail 1,000 each week for ten weeks or 2,000 a week for five weeks—whatever will help match your resources to the potential volume of incoming orders. This will stagger the response and make fulfillment much more manageable. Understanding how the response cycle ebbs and flows is also useful in avoiding abandoned calls and phone jams.

In addition, bear in mind that the response rate of up to 13 to 18 percent that we've seen as being typical to upgrade campaigns is higher than the average response to other direct-marketing offers (which run from half a percent to three percent). So if your

company is already doing other direct-marketing business, an upgrade campaign will cause a major spike in activity during that time.

ORDER-ENTRY MISTAKES

Entry mistakes are expensive and time-consuming to correct. For example, if you incorrectly enter even a single digit of a credit card number, correcting that error costs dearly in labor, bank charges, customer callbacks, and customer ill-will and loss of confidence.

Conducting telephone order entry therefore requires a specific skill: typing without looking at the keyboard. People who must do otherwise are the single largest cause of order-entry mistakes. If you must look at the keyboard, it's very difficult to type an order correctly while carrying on an intelligent conversation with a customer. So it's important that order takers have this special skill. And of course, because these people also answer the phones, they must be articulate and well-informed, understand and use good phone etiquette, express themselves well on the phone, and so forth.

INFORMATION REQUESTS

Only 50 to 60 percent of people actually place an order the first time they call. Many of these nonbuyers want more detailed information such as spec sheets, demo disks, and answers to technical questions. Closing these sales depends on how well you are able to supply that information. You must hire and train people who are capable of answering questions as well as taking orders, and you should be ready to fulfill information requests immediately. This means keeping at hand a well-replenished stock of whatever literature is appropriate and having a system in place to quickly send all needed information to a caller.

You may opt to have the same operators who take orders also answer questions, or you can plan a strategy for forwarding the caller to an appropriate designated support person. In any case, the process should be effortless and as transparent as possible to the caller.

CREDIT CARD AND CHECK PROCESSING, OPEN ORDERS

The majority of customers—60 to 80 percent—pay for upgrades with a credit card. Here the bottleneck is getting the card approved, which can be a very time-consuming process that can be complicated by invalid accounts and errors in recording the card number and expiration date.

Handling hundreds of small checks is another unbelievably people-intensive process, because of the sheer volume and attention to detail required. As many as 5 to 10 percent of checks bounce, which can create serious problems for a developer.

Another thing that can clog the pipes is the process of matching checks to the original orders. Often, a check comes from a different office than the one the user listed, and it sometimes carries another company name. If an order number isn't noted on a customer's check, the matching process can be a showstopper.

Anticipating this problem and putting procedures in place to handle the flow of checks and credit cards will help avoid this potential jam.

Open Orders. When customers offer purchase orders or checks to pay for upgrades, most developers put the order on hold until the money actually arrives. As many as 35 percent of these payments will never show up without prodding, even though the customer was really prepared to pay. To capture the enormous amount of money left

on the table, open orders should be pursued aggressively with invoices and follow-up calls.

SINGLE-COPY FULFILLMENT, ORDER TRACKING

Most software companies are used to shipping pallets or truckloads of products to a handful of large accounts. It's suddenly very different when you have to ship tens of thousands of "onesies." An inefficient shipping system can make fulfillment costs skyrocket; in addition, upgrade customers tend to judge software companies by their turnaround time. People don't want to wait three or four weeks for your product, when another company ships its goods immediately.

Setting up shipping procedures to handle this influx of small orders goes a long way toward being able to fill customer orders in a timely fashion.

Order Tracking. Especially when shipping is delayed, a lot of customers—about 30 percent—call back to find out what happened to their orders. From their point of view, you're just waiting for their calls and are prepared to ship the product the moment you hang up the phone. You need to be able to give the customer an acceptable response. For instance, are you still waiting for the customer's purchase order or check? Is the customer located in a shipping zone that takes five days for delivery, and you shipped the product only two days ago? Did the product actually ship?

If you don't have an automated order-tracking system, you're going to spend a lot of time and money making those customers happy. Customer service in general, and in particular your ability to give customers answers, is probably the Number 1 determinant of whether they will do business with you in the future.

RETURNS

As many as five percent of upgrade customers will take advantage of a money-back guarantee (or will discover that they ordered the wrong disk format or have a similar problem). It's amazing how many people will call to complain that they didn't get what they wanted.

Getting prompt refunds to these customers is important, in part because credit card companies may impose as much as a \$12 fee for charge-backs that involve a customer complaint. A refund process that isn't simple and quick (for you and the customers) can be a real bottleneck in your campaign and can stall your relationship with those customers forever.

PUT YOUR USERS INTO THE LATEST TECHNOLOGY

Handling the upgrade process correctly is not easy, but it certainly is essential to the long-term viability of your business. How you address this business segment and what level of importance you assign to it in your sales and marketing plans will have a direct bearing on the future of your company. And, as we often tell our clients, "If you don't put your users into the latest technology, someone else will!"

Jordan Levy is president of Upgrade Corporation of America, a Buffalo, New York, company that specializes in software-upgrade marketing.

Portions of this article were reprinted with permission from the Sept. 4, 1991 issue of Soft•Letter, Watertown, Massachusetts.

Hiring a Campaign Manager

If the prospect of handling an upgrade campaign on your own is daunting, you might consider hiring an outside firm to do part or all of the job. When evaluating companies, here are some things you should look for:

Scope of Services. Whether you need a full-blown, turnkey operation with the entire range of services or a company to handle a smaller piece of the action, you should match your needs to the services offered. For example, can the company do only inbound order taking or can it also do outbound telemarketing? Can it warehouse and send literature? Can it follow up with a customer after sending the literature? To what extent does it get involved in customer service?

To help evaluate this, it might be useful to create a “shopping list” of services you might need. Make it as explicit and complete as possible. Then use that list—along with what you’ve learned from talking to those companies and to other developers who have experience with upgrade fulfillment companies—to find the right match of your needs to fulfillment options. (You may also hire an experienced professional to help you define your needs.)

Industry Knowledge. Does the company have experience and expertise in the software industry? This can be particularly important, because people who buy software naturally have different information needs than people who buy kitchen knives or music. Software customers who call for information don’t just want to differentiate between the serrated-edge blade and the smooth one or between

buying the cassette or the CD; they have more sophisticated information needs, such as “Will this package run on my configuration or on my network?” The quality of the answers they get can have a direct impact on whether they buy your product.

Credentials and References. References are very important. Getting an endorsement (or a warning) from a colleague who has used a particular company will help you gauge whether what that company says about itself is actually true. It can also help you better understand a company’s strengths and weaknesses from a developer’s point of view.

Also, if you can tap colleagues for information that will help you avoid pitfalls or define your needs, all the better. You’d be surprised how helpful people can be who have been down a road before you.

Pricing. Getting a handle on pricing is one of the most difficult aspects of evaluating a fulfillment company. It is sometimes very hard to determine, up front, exactly how much a fulfillment project will cost. One of the best ways to compare pricing among companies is to combine your list of needed services with your best estimate of the volume of business, and give it to each company you are evaluating.

Since fulfillment companies each have their own way of packaging and pricing “basic” services and pricing individual ones, it may still be difficult to compare the cost estimates you get— but this exercise will be helpful.

You should know up front that everything, *everything* a fulfillment company does carries a price tag; make sure that you understand the company’s pricing strategy, learn exactly what is included in the pricing, and make sure that every activity you might need is defined and that a price is affixed to it.

Two things that often influence pricing are the volume of your business and the longevity of the contract. Generally speaking, a one-time project of 600 units (a small

volume) will cost significantly more per unit handled than a longer-term contract with a higher volume of work. There are companies that specialize in smaller-volume jobs; if your volume is low or you are “piecing out” only a part of the fulfillment process and don’t need a more complete turnkey operation, smaller fulfillment companies may be good choices.

(Somewhat) Standard Costs. As you examine pricing, it’s important to note, that out-of-pocket items should be similarly priced from vendor to vendor:

- Credit card fees. This is the percentage a credit card company charges a merchant for each transaction. Fees run the gamut, but the average in the U.S. is 3.6 percent.
- Cost of shipping the product, literature, and so forth. No matter which shipping service you use, the cost for that service is set by the supplier and is consistent for all users of that service. To give you an idea of what shipping can cost, a four-pound package costs about \$3.36 to ship via UPS; that figure is based on the average cost of shipping that package to and from all UPS zones.
- Cost of the phone call needed to take orders. Again, those rates are set by the supplier and should vary only by which supplier a fulfillment company uses. The average cost is 20 cents per minute, and the average inbound call to place an order lasts four minutes.

A Pricing Strategy. To give you an idea of how fulfillment pricing is built, here’s an example:

Step 1: Start with your four-pound product that sells for \$100 plus a \$10 shipping-and-handling fee. That makes your price \$110. Now add sales tax. If you average the sales taxes from two states (your state plus the one in which the fulfillment company is located), you’ll find that it averages approximately 2 percent across the U. S. $\$110 \times .02 = \2.20 . Add that to your product cost, which makes \$112.20.

Now start adding fulfillment costs:

Step 2: For simplicity's sake, let's assume that you don't accept purchase orders. Eighty percent of orders will be paid by credit card, and the credit card fees will average 3.6 percent of the order. $\$112.20 \times .036 \times .8 = \3.23 .

Step 3: The average UPS charge for a four-pound package shipped by ground is \$3.36.

Step 4: The cost of the average phone call (four minutes long) to place an order is 80 cents.

Step 5: You'll need a shipping container to protect the product en route. A plain brown corrugated shipping box will probably cost at least 25 cents (this cost is quantity sensitive). If your box is fancier, it will be more expensive.

At this point, fulfillment has already cost you \$7.64 (add the costs incurred in steps 2 through 5), the out-of-pocket expenses—and we haven't yet calculated labor, service, and a reasonable profit for the fulfillment company. What if you had to ship a piece of literature to a prospective customer? There's the four-minute average phone call at 80 cents; plus the salary of the person taking the order; plus the envelope or box; plus postage, warehousing, profit for the fulfillment company, and so forth...

I hope this gives you a basis for understanding how fulfillment pricing can be built, and that this will help you ask the right pricing questions when evaluating a fulfillment house.

-end-

Compatibility Testing

The right way, the right time

by Mike Taber and Karyn Taylor, Bröderbund Software

Not too long ago, we had a banner in the quality assurance lab that read, “If it boots, ship it!” Obviously this was intended as a joke, but the sad fact is that developers often feel pressure to get products out the door on schedule, sometimes at the expense of adequate compatibility testing.

Of course, doing this can eventually have a dramatic impact on your company’s bottom line. Although some people might feel that the “shop” or “lab” activities related to producing a product (quality testing, compatibility testing, and the like) are far removed from the process of marketing a product to customers, they couldn’t not be more mistaken.

One of the hottest issues in any industry these days is Quality. And nowhere is this more important than in the software industry. In the past, users have been pretty tolerant of software errors, (almost) affectionately known as “bugs” by both developers and users. But as people increasingly demand both ease of use and a high level of productivity from their computers, they are becoming much less tolerant of unreliable software. How well you’ve tested your product for compatibility can directly influence customer satisfaction and will ultimately determine whether the product continues to sell.

Of course, developers want to produce reliable software. But achieving this is another matter entirely. Software testing is a relatively new and somewhat unproven “science.” It is an amorphous and sometimes elusive mix of art, science, and lots of hard work.

But one thing is for sure: Testing to ensure software quality is labor-intensive and time-consuming. On the average, testing accounts for 50 percent of the total software development effort. And let’s face it, compatibility testing is probably the most tedious aspect of the entire testing process. With a tight production schedule, the pressure to squeeze testing into fewer and shorter cycles is enormous. Not surprisingly, it’s the compatibility side of the equation that often gets the short end of the stick. But failure

to fully test for compatibility—or putting compatibility testing off to the last minute—is more and more a high-stakes gamble.

WHY TEST?

Compatibility testing typically serves two very important purposes. First, it helps ensure that a standard of quality is established and maintained from product to product. And second, it helps minimize perhaps the single largest burden of software maintenance: revisions due to incompatibility with other products.

The second issue is becoming more important all the time, because we are no longer dealing with a situation in which people basically use one or two applications and a stock hardware/software configuration.

Today, Macintosh customers use their computers for some pretty sophisticated stuff. They typically use multiple applications, numerous utilities and extensions, and a variety of other specialized applications and hardware. Developers must contend with a variety of system extensions, specialized boards and chips, and other add-ons—in addition to everything else.

Compatibility testing assures that your application will continue to function on new CPUs or under new OS versions. Users can be assured that your application will always perform to specification and won't go down in flames just because of a minor hardware or software configuration change.

And it's not enough for your program to perform well as a stand-alone application. In today's market, we can no longer afford the luxury of such naive arrogance. Every application has to be fully compatible with every other Mac program on the market, as well as with the entire Macintosh line. Customer satisfaction today clearly demands a comprehensive compatibility testing strategy.

THE IMPORTANCE OF GOOD TEST DESIGN/PLANNING

Compatibility testing—to ensure that a product will perform to spec on all the platforms and in all the various configurations for which it was intended—includes not only the running or execution of tests but also the design and planning of the testing process.

How well you plan and design the process is especially important, because compatibility testing, like the vast majority of testing, typically uses black-box testing methods in which you can't see the code (as opposed to glass-box testing, in which you have access to the code).

This particular type of testing always poses a unique challenge; you've got to be adept at uncovering problem areas and diagnosing the symptoms without really seeing what you're doing.

So to a large extent, the quality of the test design/plan shapes both the effectiveness and the efficiency of the test process as well as the quality of the final product.

Without a doubt, the test design/plan can make or break the effectiveness of testing. Run without a planned, systematic approach, tests aren't likely to pinpoint all the product's deficiencies and thus won't meet all your objectives.

Obviously, a primary objective is to locate software errors. And sure, anybody can test software (unfortunately, customers do it more often than we care to admit); that is, if all you mean by testing is to sit someone down, tell them to use the program, and report any bugs they find.

Just about everybody can find bugs this way—maybe even lots of bugs—but it's a very hit-or-miss thing. Furthermore, the chances are good that you won't have enough information to be able to successfully reproduce any of the bugs, never mind being able to isolate the problems and provide the programmer with detailed recipes for reproducing the bugs found. This is hardly good software testing. It lacks a well-planned design based on a systematic approach.

Also, an often overlooked testing objective is to ensure that the application meets (or exceeds) customer expectations. All too often, good software has a minimal number of errors and still fails to meet consumers' expectations. The test-design phase—not after the product ships—is the time to ensure that the product will meet consumers' requirements, needs, and expectations.

Furthermore, using a well-designed, systematic approach can help detect errors even before code is written. As soon as an application's specifications have been defined (early in the process), you should begin designing the types of tests to be executed and determine what your testing configurations will be.

Whenever possible, make these test strategies and configuration checklists available to the programmers along with the product's spec sheet. This way your programmers will know not only what the application is supposed to do but also what kinds of tests you will run.

This increases the likelihood that the code will be written with compatibility issues as well as specifications in mind, right from the start. Obviously, this is the least costly time to correct errors. Late detection of basic compatibility errors can be an expensive proposition. Extensive debugging and several additional cycles of testing may be needed to fix problems that could have been avoided altogether.

CONFIGURATION MANAGEMENT

In addition to a good plan and test design, effective compatibility testing also requires a good configuration-management system. A practical way to approach this is to set three basic configuration levels: the base, added functionality, and real-world levels. Simply defined, a base-level configuration is a Macintosh right out of the box. The software/hardware configuration is what Apple ships, period, with no additional system extensions, no add-on boards or chips—nothing extra whatsoever.

The added-functionality level is the base configuration plus additions to the system that enhance its functionality. This is the configuration level at which you plug in the extensions and other such things you claim (or want) to support. This is also the time to add the alternative video adapters, memory expansions, and so forth.

The real-world configuration level is just what it implies, a real-world configuration where just about anything goes. Such a configuration is typically loaded with all kinds of bells and whistles that don't necessarily enhance the functionality or ease of use of your applications, such as screen savers and other fun things that can (and will!) wreak havoc on your application if they aren't adequately tested.

The challenge at the added-functionality and real-world levels is to figure out what types of configurations your users have. There are several ways to determine this.

A good place to start is with the system on your desk at home or at work. What kinds of goodies are installed? Surely you must have a few things that always give you trouble. You can also read the latest magazines to determine which new products are getting good reviews or those that just plain sound good to you. These will undoubtedly be the items your customers buy and use. You can also use more formal market research to determine what products your customers buy.

With your configuration levels established, you must determine exactly which configurations to actually test. This is a relatively simple task; tailor your test configurations to the specifications and functionality of your application. Some tests will simply evaluate the baseline functionality of your application, and a variety of tests (for such things as file/disk I/O, edit options—Cut, Copy, Paste, and the like—printing, virtual memory, and 32-bit addressing) can be standardized and used from product to product.

LET THE TESTS BEGIN

With all the pieces in place, you're ready to test. The best time to perform compatibility tests is during regular testing. To do this, you must define your test control environment. It is useful to employ both ideal and low-end configurations as controls—what you'd ideally like to see your application run on and a worst-case scenario. This will give you two opposites to compare.

You may ask yourself why anyone would run your sophisticated state-of-the-art painting program on a monochrome Macintosh Plus. The truth is, if it can be launched on a Plus, it will be used on a Plus. The sad fact is that if your application doesn't perform adequately on a Plus, it should fail your compatibility tests.

Regression tests can also be useful in compatibility testing. Regression testing is simply rerunning a test set to verify that your application's functionality and performance are in accordance with specifications.

They are traditionally run every time a product change is made, whether it be to the application's code, requirements, or system configuration. Which test sets to rerun is determined by the type of change

introduced. Once changes are implemented, your regression tests will be run over and over and will assist in establishing a baseline for the testing that will help avoid a runaway test process.

Here's an example: Let's say that a new CPU is being developed; your application needs to perform to specification on it. Here's where your regression test sets come in handy. Simply run them on the new CPU in various configurations.

These regression tests may be no more than simple function tests; an even better test set might be to give the CPU "the works"—a variation on classic stress testing. Create a set of test suites that pits your application's most complex features (disk- or memory-intensive operations) against the new hardware. If your application can pass these tests, chances are good that its performance will hold up across the board in the new hardware environment.

RESOURCES AND SERVICES AT YOUR DISPOSAL

A variety of resources and services is available to developers to complement the compatibility testing process. We highly recommend that you take advantage of these. They include user groups, service bureaus, and of course, Apple's Third-Party Compatibility Test Lab.

Apple's facility gives developers access to a wide range of Macintosh configurations and networking environments. Your testing team will be able to test its products on systems that have been preconfigured to your specification—at no charge to you.

Also, if during your testing at the lab you encounter errors in your software, the Apple engineering-support team will be available to assist you in debugging it. Time and time again, our experiences have proven the Third-Party Compatibility Test Lab to be an invaluable resource in our overall compatibility testing process.

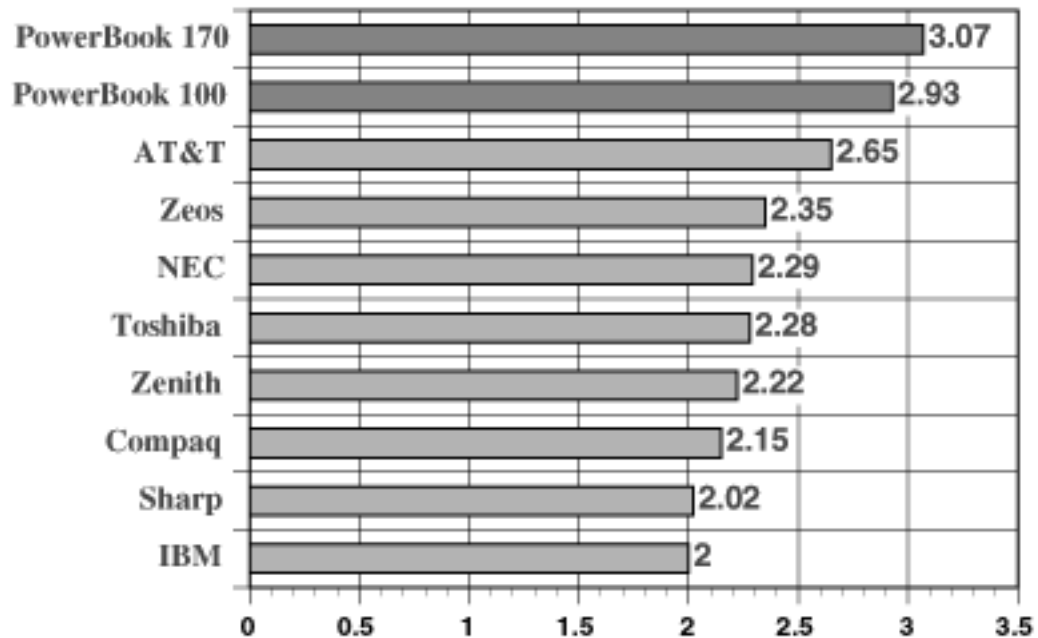
One point we have tried to emphasize throughout this discussion is that, like it or not, compatibility testing is a never-ending process. And because it has an impact on your product's overall success, we strongly urge you to take the time and effort to do it well. See you at the lab!

Mike Taber is a senior lead quality-assurance technician, and Karyn Taylor is a lead quality-assurance technician at Bröderbund Software, Inc., an education and consumer software development company located in Novato, California.

Marketing Folder on the CD and AppleLink

Apple's monthly Developer CD now also features material for nontechnical types, namely a marketing folder with resources, tips, data, and other marketing-related information. It includes a folder containing all the *Apple Direct* marketing articles that have appeared in the last year and a half as well as other folders containing international market guides, new-product data sheets, a U.S. distribution guide, and product-training information. You'll find the marketing folder at the top level of the CD. There's also a marketing folder on AppleLink (path—Developer Support:Developer Services:Marketing) offering the same types of information.

Apple's PowerBooks Rated Most Usable in Independent Tests



GVO, an independent industrial-design company, recently put ten prominent notebook computers—from companies including IBM, Compaq, Toshiba, NEC, Zenith, and Apple—through more than 450 real-world usability tests. The tests were performed across the whole notebook-computing experience—from the time the user first opens the box through extended use in the office and in the field—and included typical customer situations such as storing the computer in a briefcase and using it on an airplane and in a car. The Macintosh PowerBook models 170 and 100 were rated number 1 and 2 in the tests (the PowerBook 140 wasn't tested, because of its similarity to the 170). If you'd like to read a full report describing the entire study and GVO's conclusions, look on this month's Developer CD in the "What's New on This CD?" folder.

Source: GVO, 1991

APPLE DIRECT

Apple Direct is Apple's monthly developer newspaper, covering business and technical issues for decision makers at 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, Todd Luchette, Kris Newby, Karen Wickre

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, HyperCard, LaserWriter, LocalTalk, Macintosh, MacApp, MacTCP, MPW, SANE, and SADE are trademarks of Apple Computer, Inc., registered in the U.S. and other countries. Balloon Help, develop, 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. UNIX is a registered trademark of UNIX Systems Laboratories.

Mention of products in this newspaper is for informational purposes only and constitutes neither an endorsement nor a recommendation. All product specifications and descriptions were supplied by the respective vendor or supplier. Apple assumes no responsibility with regard to the selection, performance, or use of the products listed in this newspaper. All understandings, agreements, or warranties take place directly between the vendors and prospective users. Limitation of liability: Apple makes no warranties with respect to the contents of products listed in this newspaper or of the

completeness or accuracy of this publication. Apple specifically disclaims all warranties, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.