



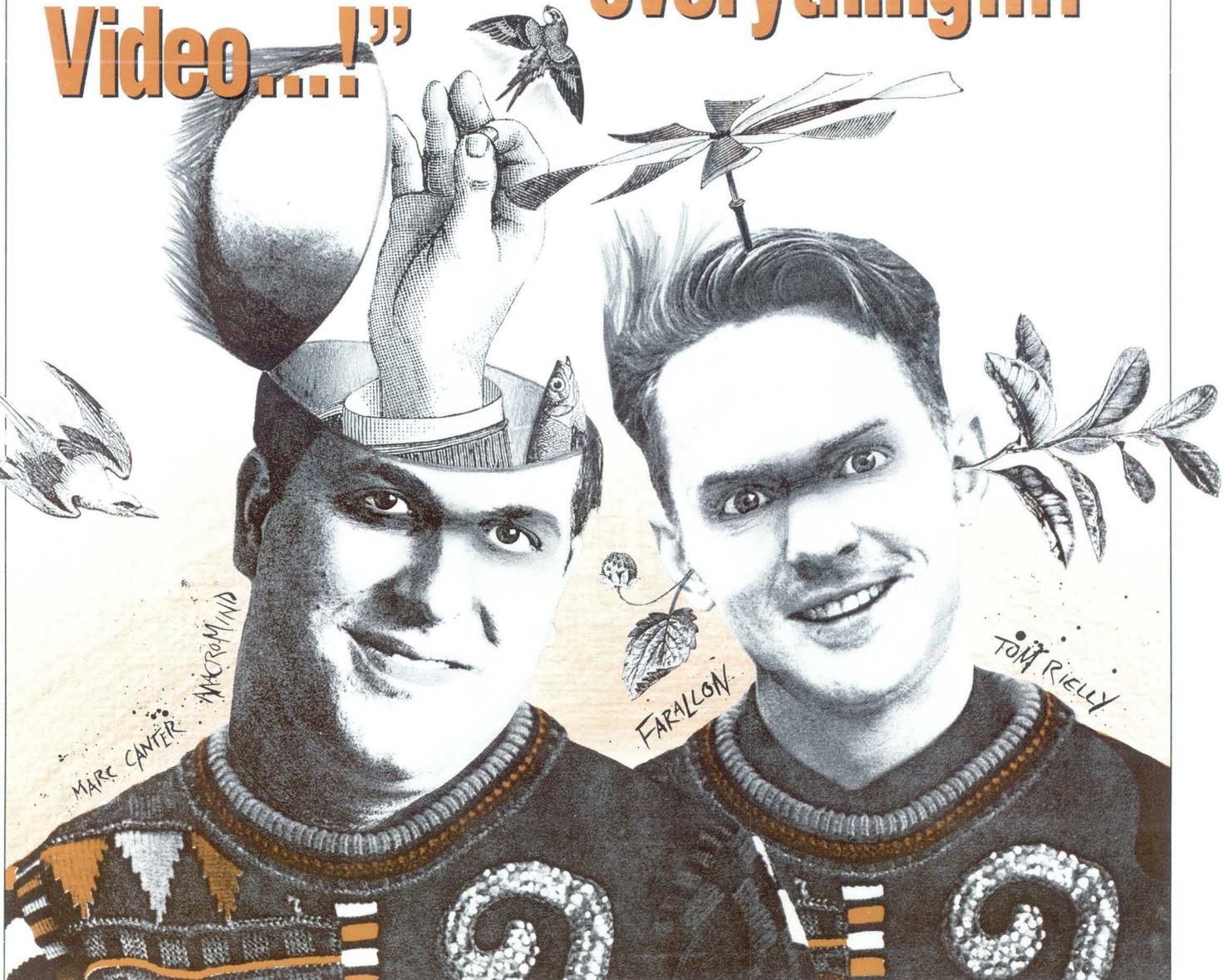
Apple *direct*

VOL. 2, NO.10

AN INFORMATION RESOURCE FROM APPLE DEVELOPER SERVICES • JULY 1990

**“Sound.
Animation.
Video...!”**

**“This changes
everything...!”**



July 1990

Features

8/ Changing Everything with Media Integration

By Gregg Williams and Lisa Raleigh

Sound, video, animation, and more...

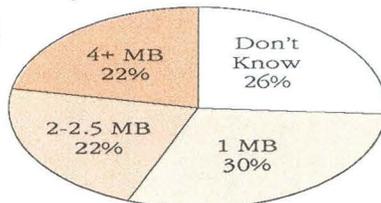
(Just in case you didn't recognize them, our cover models are Marc Canter, Chairman of MacroMind, and Tom Rielly, director of marketing at Farallon.)



15/ Apple Surveys Its Customers

By Lisa Raleigh

Many are 7.0-capable.



© 1990 Apple Computer, Inc. Apple, the Apple logo, Apple IIgs, AppleLink, AppleShare, AppleTalk, GS/OS, HyperCard, IIgs, LaserWriter, LocalTalk, Mac, MacApp, Macintosh, MacTCE, MPW, ProDOS, Reanimator and SANE are registered trademarks of Apple Computer, Inc. APDA, APDalog, APW, GS/Bug, MacAPPC, MacinTalk, MacWorkstation, ResEdit, Ram Dump, QuickDraw, and TrueType are trademarks of Apple Computer, Inc. Apple Works is a registered trademark of Apple Computer Inc., licensed to Claris Corporation. APPMaker and APPMaker GS are trademarks of Bowers Development Corporation. Design Master is a trademark of Byte Works. Full Impact is a trademark of Ashton-Tate, Inc. Genesys is a trademark of SSSI, Inc. HyperAnimator and InterFACE are trademarks of Bright Star Technology. Microsoft, MS-DOS and Windows are trademarks of Microsoft Corporation. MacroMind Director and MacroMind Multi-Media Manager are trademarks of MacroMind, Inc. MacRecorder and MediaTracks are trademarks of Farallon Computing. Mitem is a trademark of Mitem Corporation. OS/2 and Presentation Manager are trademarks of IBM. PixelPaint is a trademark of SuperMac Technologies, Inc. QuickMail is a trademark of CE Software, Inc. Voice Link is a trademark of Articulate Systems, Inc. Allegro Common Lisp is a trademark of Franz, Inc. LaserJet is a trademark of Hewlett-Packard Company. Diablo is a trademark of Xerox Corporation. QuickMail is a trademark of CE Software, Inc. SuperCard is a trademark of Silicon Beach Software, Inc. Think C is a trademark of Symantec Corporation. NuBus is a trademark of Texas Instruments. Presentation Manager is a trademark of International Business Machines Corporation. PostScript is a registered trademark of Adobe Systems Incorporated. WordPerfect is a registered trademark of WordPerfect Corporation. Motorola is a registered trademark of Motorola Corporation. OS/2 is a registered trademark of International Business Machines Corporation.

Departments

3/ EDITOR'S NOTES

By Bill Freais
The Apple II developers' conference

4/ FYI

By the Apple Direct Staff
New LaserWriters, compatibility alerts, a cache card, and more

6/ LATEST REV

A listing of the latest version numbers of Apple's system software

17/ ELECTRONIC MEDIA

By Jessa Vartanian and Scott Converse
System 7.0, training tools, and XCMDs



19/ THE HUMAN INTERFACE

By Bruce Tognazzini
Is the Macintosh interface intuitive and consistent?

21/ DEAR DEVELOPER

By Bill Hollifield
I hate "Install."

22/ IT SHIPPED

A listing of third-party products that shipped between April 27 and May 31

23/ MARKET INSIGHT

By Jim Davis
Seeing through Windows

26/ VIEWPOINT

By Jim Merritt
Parallel development: learn while you earn

28/ GETNEXTEVENT

A listing of upcoming events pertinent to developers



Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, California 95014
(408) 996-1010
TLX 171-576

Editor's Notes

The Apple II Developers' Conference



AS THIS ISSUE OF *APPLE DIRECT* WENDS ITS WAY TO YOU, 150 OR SO Apple® II developers and Apple employees are heading for a long weekend on the Avila College campus in Kansas City, Missouri—city of fountains, the Royals, and the Second Annual A-2 Central Summer Conference (known otherwise as the Apple II Developers' Conference or KansasFest). For two days, developers will focus on state-of-the-art Apple II technical issues to the exclusion of all other worldly concerns (except, that is, Saturday night's Royals/Red Sox game, included in the \$300 admission price).

Apple's Apple II Developer Technical Support team will be there in force. Over the course of the weekend, engineers from Apple II System Software and Apple II DTS will re-enact their contribution to Apple's Worldwide Developers Conference, held last May at the San Jose Convention Center here in California. Among other topics, their sessions will cover IIGs® system software, graphics and animation techniques, sound tools, programming strategies, media integration, and parallel development. There will also be a discussion of Apple II system software and development tools.

And that's not all from Developer Technical Support—the day before the conference officially begins, DTS engineers will conduct the second Apple IIGs College (the first IIGs College was held in spring of 1987, when the IIGs was introduced). There, for an extra \$100, Apple II developers can get the bare essentials on the Apple IIGs operating system and tools. They'll also get programming strategies, debugging techniques, and tips on segmentation (the magic of squeezing now-sizable Apple IIGs applications into 1MB). Toward the end of the day, attendees will choose a programming project and attempt to pull it off under the supervision of DTS engineers (yes, Apple's devised more sinister tortures, but no one's lived to talk about them).

The Apple II Product Development team will also participate in the conference. Mark Cecys, who recently developed a sound tool for the IIGs, will be there, as will Tim Swihart, who'll discuss Apple II programming in the '90s. And for those who find their interest in engineering topics waning, David Szetela, Developer Services' point man, and Jane Lee, the Apple II Marketing Manager, will be there to give Apple's perspective on Apple II development.

But lest you think Apple's taken the A-2 Central Summer Conference over completely, you should know that many independent Apple II developers will also be giving presentations. In fact, there'll be more than 30 technical sessions in addition to what Apple people are giving. Among the non-Apple presenters will be Roger Wagner, Randy Brandt, Morgan Davis, and Barney Stone.

Developers are constantly wondering about Apple's position vis-à-vis the Apple II. Although *Apple Direct* can't predict what the future holds for the Apple II, we can tell you this: The Apple II is a valuable and robust product, and as such it still presents plenty of development opportunities. There are more than 5 million of them out there, primarily in the schools, which means lots of people have a vested interest in the well-being of this machine and its third-party software. As long as that's the case, Apple will be there to support it. 🍏

Bill Freais, Departments Editor

Appledirect

PUBLICATIONS TEAM

EDITOR
LISA RALEIGH

TECHNICAL WRITER/EDITOR
GREGG WILLIAMS

DEPARTMENTS EDITOR
BILL FREAIS

VIEWPOINTS EDITOR
DEE KIAMY

PRODUCTION
COORDINATOR
DIANE WILCOX

DESIGN / PRODUCTION
FISHER & DAY

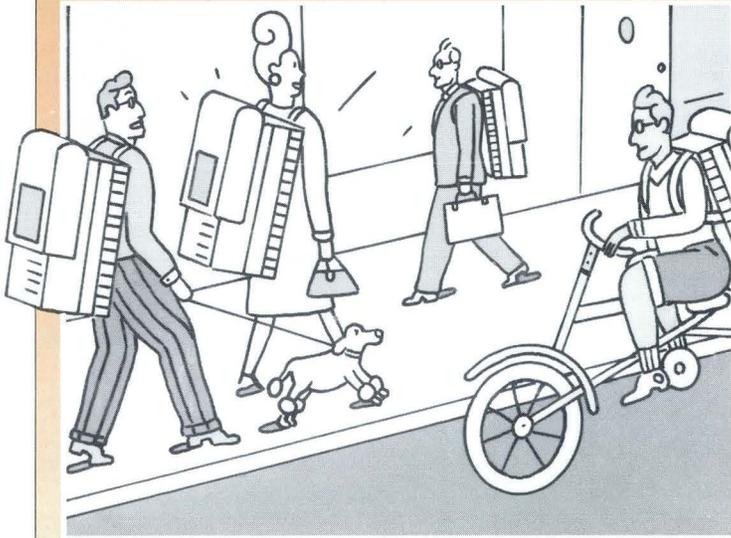
MANAGER DEVELOPER PRESS
DAVID KRATHWOHL

Apple Direct is published monthly by Apple Computer Inc.'s Developer Press group, part of the Developer Services organization.

MENTION OF PRODUCTS IN THIS newsletter is for informational purposes only and constitutes neither an endorsement or 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 newsletter. 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 newsletter, or 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. *FYI illustrations by Stephen Turk. Cover Story illustrations by Craig Hansen. Editor's photo by Ron Grant Tussy. Cover illustration by Fisher & Day.*

FYI



LaserWriters for the Rest Of Us

BY INTRODUCING THE Personal LaserWriter® SC and Personal LaserWriter NT printers earlier this month, Apple made good on its promise to bring more-affordable LaserWriter-quality printing to Macintosh® customers.

The Personal LaserWriter SC is designed for individual use; the NT is designed for individual as well as group use. Both give you full-page, high-resolution printing for a wide variety of documents—from letters and memos to full pages of sophisticated text and graphics that include scanned illustrations and photographs, page layouts, and presentations. And both can deliver up to 4 pages per minute; they're rated at 150,000 pages—which means 100 printed pages per day, 5 days a week, 52 weeks a year, for more than 5 years.

The Personal LaserWriter SC comes with a selection of bit-

mapped fonts—Courier, Helvetica, Symbol, and Times—but it can also support additional fonts, as well as the Adobe Type Manager. When System 7.0 becomes available later this year, the SC will be compatible with TrueType™ fonts.

With the Personal LaserWriter NT, you get 12 PostScript® font families that can be printed in a virtually unlimited range of sizes and styles: ITC Avant Garde, ITC Bookman, Courier, Helvetica, Helvetica Narrow, New Century Schoolbook, Palatino, Symbol, Times, ITC Zapf Chancery, ITC Zapf Dingbats, and IBM PC Graphics Extended Character Set (ECS). The NT also supports hundreds of other fonts available from a variety of companies.

A particular advantage of the the Personal LaserWriter NT is its ability to operate in a network of Macintosh as well as MS-DOS computers. LocalTalk® networking capabilities are built in to the LaserWriter NT, so connecting it to a single Macintosh or to a network simply involves plugging in the LocalTalk cable. MS-DOS

computers can also easily print to the Personal LaserWriter NT. To print via LocalTalk, all that's needed is a LocalTalk PC peripheral card. The Personal LaserWriter NT printer comes with an RS-232 interface, which allows anyone whose computer supports this interface to use the Personal LaserWriter NT to print high-quality PostScript text and graphics—or to print non-PostScript documents, using HP LaserJet Plus or Diablo 630 printer emulation.

And note: The SC can be upgraded to a Personal LaserWriter NT by a simple upgrade of the controller board. 🍏

More Cache for Your IICI

FOR THOSE OF YOU who regard super high performance as a staple, Apple's also just announced a cache card for the Macintosh IICI. It features 32K of high-speed static RAM and a custom cache-controller ASIC mounted on a 3-inch-by-6.1-inch printed circuit board, all of which snaps easily into the 120-pin EuroDIN-style cache connector on the IICI's logic board.

The IICI cache card is essentially a performance booster and will appeal most to those working with financial spreadsheets, graphics, desktop publishing, CAD/CAM, and engineering and scientific applications. By conservative estimates, the card increases the IICI's performance by 20 to 30 percent. 🍏

Future CPU Compatibility Alert

AS APPLE FOCUSES ON EXPANDING its product line and making Macintosh computers even more accessible, you should be all the more aware of compatibility issues. We'll do our best to keep you apprised of these issues as they come to our attention. For now, however, note the following compatibility problems you might experience with our new CPUs and color monitors.

Future Macintosh CPUs will have optional user-installable floating-point units (FPUs). The key word here is *optional*: your application should not assume that every Macintosh with a microprocessor greater than a 68000 has a Motorola 68881 or 68882 FPU. The only way to ensure that your application will remain compatible with future Macintosh computers is to use the Gestalt Manager (the successor to Sys Environments) to determine the exact configuration of the machine you are running on. This means that if you offer two versions of your application, one that uses SANE® (Standard Apple Numeric Environment) software and another that does its numeric calculations using FPU hardware, or if your application makes a conditional branch to execute floating-point instructions directly, then the application should first check for the presence of the FPU. The following pseudocode shows the right way and the wrong way to do things.

Wrong:

```
IF I require an FPU THEN
  IF Microprocessor > 68000 THEN
    FPU_Present = True
  END IF
END IF
```

Right:

```
IF I require an FPU THEN
  Call Gestalt Manager
  IF FPU is present THEN
    FPU_Present = True
  END IF
END IF
```

Also, your application shouldn't assume that all color monitors have a resolution of 72 dots per inch or minimum dimensions of 640 by 480 pixels. Your application should use the Gestalt Manager to determine the exact resolution and dimension of the monitor the user has configured. The Gestalt Manager has been included in System software since 6.0.4 and is described in your System 7.0 documentation. Remember, for your application to be compatible with all new CPUs, it should explicitly check for every feature it requires. 🍏

Keep Those Cards Compatible

AND WHILE WE'RE ON THE subject of compatibility, a preliminary release of the *SE/30 Preliminary Notes* gave contradictory information about a pinout and a chart for the SE/30 processor direct slot (PDS). The CPUCLK (the CPU clock signal) on the SE/30 PDS is pin A38, and C16M (a 16MHz clock signal) is

pin C38—not the other way around. If you develop SE/30 cards, you should determine where your card takes its clock signal. If your SE/30 card expects C16M on pin A38 or CPUCLK on pin C38, then it will most likely not be compatible on future CPUs that use the SE/30 PDS. As always, Apple recommends that you design your cards to be clock-speed-independent.

SIAC, Apple's System Integration and Application Compatibility group, is creating a library that will help increase the probability that third-party hardware sent to us will be examined for compatibility. To increase the probability that *your* expansion



cards will meet the compatibility test, we urge you to donate one or more of each of them as soon as possible to Apple's Engineering Support group. Your cards may be tested for form-factor and functional compatibility (when possible) with new Macintosh CPUs and peripherals.

Apple does not guarantee that your card will be tested if it is donated, but we can promise that it will not be tested if it isn't.

We currently need SE/30 and NuBus cards. If you can't afford to donate your card(s), you can arrange to lend them to us (and you'll need to indicate clearly, in a transmittal letter, that

you are *lending* your card). Three months is the suggested minimum loan time. All due care will be taken in handling the cards, but Apple cannot accept responsibility for their return condition.

Mail your finished product to: Apple Computer, Inc., Engineering Support, 20450 Stevens Creek Blvd., MS 76-1E, Cupertino, CA 95014.

Be sure to complete an "It Shipped" form for each product you submit. Engineering Support will use this form to enter data about your cards into its database. The form is available on AppleLink® (Developer Services:Developer Programs:It Shipped! Program) and can be obtained from the Developer Hotline at (408) 974-4897. 🍏

Dow Jones' Developer Program

THESE DAYS, THE word information strikes fear into the heart. Not because information's hard to get, but because it's hard to manage. Raw information abounds; the trick is being able to sift through the abundance to find what you need when you need it.

Recently, Dow Jones, one of the largest purveyors of information in the world, devised an innovative way to meet today's information needs, and by so doing has provided Macintosh developers with an interesting opportunity. Dow Jones calls this opportunity DowVision, and it is

comprised of five parts: proprietary news from Dow Jones and information from other sources, a network delivery system, a broadcast server, an interactive server, and the Alliance Developer program.

DowVision's broadcast and interactive servers offer a vast array of information sources. The broadcast server itself merges several news wires into a single integrated, highly structured stream of information, flowing over a dedicated, high-speed, X.25 packet-switching network. Within that stream, customers will find the family of news wires called *Dow Jones International News Service*, as well as *PR Newswire* and *Business Wire*. They'll also find *The Dow Jones News Service*, *Capital Markets Report*, and the *Professional Investor Report*. The broadcast feed contains extensive coding for every story—indicating subject matter, companies mentioned, geographical region, date, time, and the like—so that customers can filter, select, and prioritize stories according to criteria they determine.

Through the interactive server, customers can access delayed and real-time market data; brokerage-house reports; earnings estimates; financial profiles on public and private companies; as well as the full text of articles from more than 400 newspapers, magazines, and business publications, including *The Wall Street Journal*. Both sources of information are available for a low fixed monthly fee.

So why are we telling you about this? Because of DowVision's Alliance Developer program. There's one critical piece

FYI

missing in all this, and that's the link between customers and DowVision's galaxy of information. To establish that link, Dow Jones is offering to provide technical support, comarketing opportunities, and revenue sharing to Macintosh developers embarking on the creation of DowVision-based products that will enable Dow Jones' customers to navigate its information service transparently.

As Apple sees it, DowVision provides an excellent opportunity for Macintosh developers to incorporate an information component into existing applications or to create new types of information products, such as personalized, dynamic "newspapers."

Apple's Evangelists are working closely with Dow Jones to make DowVision a premier source of business information. If you'd like to find out more about the opportunities DowVision presents, contact Greg Gerdy, Dow Jones Advanced Systems Group, at (609) 520-4393. 🍏

Oops

IN THIS CASE, OOPS HAS nothing to do with programming environments. Rather, we'd like to set the record straight on a couple of items that appeared in March and April's *Apple Direct*.

In our April News Clips, we told you that Program Slicer/Dicer (S/D) was developed here at Apple (see "Developers Go MADA"). Well, that was wrong, and Edward J. Fremouw of NorthWest Research Associates was good enough to point out

the error. In fact, Program Slicer/Dicer was developed by Visualogic, Inc., of Bellevue, Washington, a subsidiary of NorthWest Research Associates. Its primary author was Dr. R. David Lucas.

Now for March's corrections: Many of you read about Apple's new SCSI card in last March's issue ("The Apple II High Speed SCSI Card," March 1990, page 19). Although the article is correct, overall, in describing the card, its advantages, and how it works, a few details were off. Here's the correct information:

The speed of the Apple II High Speed SCSI Card is five to ten times faster when it is using DMA than when it is using pseudo-DMA, but the 500K-per-second transfer rate for pseudo-DMA is incorrect. The upper limit of SCSI DMA transfer is 1 megabyte per second, and the card's actual performance is influenced by the speed of the SCSI peripherals used.

Again, depending on the speed of the SCSI hard-disk drive used, AppleWorks® can be loaded up to 40 percent faster with the Apple II High Speed SCSI Card, as compared with Apple's previous SCSI card; the improvement is due largely to the SCSI DMA mode. ProDOS® 8 boots up slightly faster than before. (The article incorrectly said two to three times faster for AppleWorks and *no* faster for ProDOS 8.)

As for GS/OS® driver software, users will get drivers for hard-disk drives and for CD-ROM drives; developers will get those drivers and the drivers for the Apple SCSI Scanner and the Apple SCSI Tape Drive. Apple does not have any kind of Apple

LaserWriter driver for Apple II-family computers, but they have always been able to print to LaserWriter IINT and IINTX printers over an AppleTalk® network.

Finally, our description of the Backup II program (which

said that it now handles extended GS/OS files with data forks) should be amended to say that it now correctly handles extended GS/OS files, which can contain data forks as well as resource forks. 🍏

Latest Rev, July 1990

APDA
AppleLink
Developer
CD-ROM

Apple II Family Products:

- Apple II System Disk 3.1
- APW™: Apple IIcs® Programmer's Workshop 1.0.2
- APW C: Apple IIcs Programmer's Workshop 1.0.2
- APW Tools and Interfaces 1.1
- Apple IIcs BASIC 1.0b4
- GS/Bug™ and Debugging Tools 4.0b1
- Apple IIcs Icon Editor 1.1
- Apple IIcs System Disk 5.0.2
- MPW® IIcs Tools 1.1
- MPW IIcs Assembler 1.0
- MPW IIcs C 1.0.1
- MPW IIcs Pascal 1.0b1

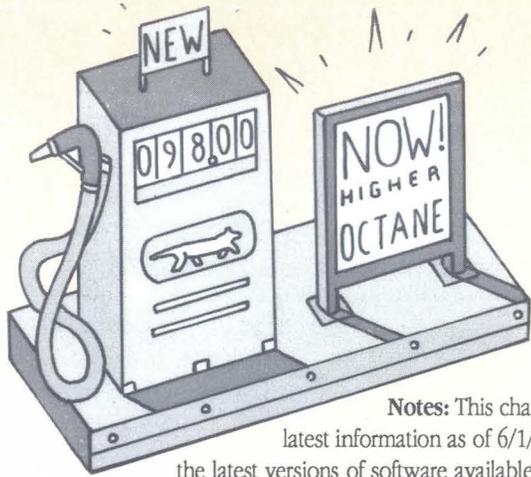
Macintosh Family Products:

- MPW: Macintosh Programmer's Workshop Development Environment 3.1
- MPW Assembler 3.1
- MPW C 3.1
- MPW C++ 3.1b1
- MPW Object Pascal 3.1
- MPW Toolbox Interfaces and Libraries 3.1
- SADE: Symbolic Application Debugging Environment 3.1
- MacApp® 2.0
- Macintosh Allegro Common Lisp™ 1.3.2
- MacsBug 6.1
- MacsBug (pre-release) 6.2a5
- ResEdit™ 1.2
- ResEdit (pre-release) 2.0b2
- MIDI Management Tools 1.2
- RAMDump™ and ReAnimator® 1.0
- RAMDump and ReAnimator (pre-release) 1.1b1

Macintosh System Software:

- Macintosh System Software 6.0.5
- Macintosh System Software—32-Bit QuickDraw™

APDA	AppleLink	Developer CD-ROM	
6.0.3		•	Macintosh System Software Arabic 6.0.4
6.0.3		•	Macintosh System Software British 6.0.5
6.0.3		•	Macintosh System Software Danish 6.0.5
6.0.3		•	Macintosh System Software Dutch/Flemish 6.0.4
•		•	Macintosh System Software Færoese 6.0.3
6.0.3		•	Macintosh System Software Finnish 6.0.5
6.0.3		•	Macintosh System Software French 6.0.5
			Macintosh System Software
6.0.3		•	French Canadian 6.0.5
6.0.3		•	Macintosh System Software German 6.0.5
6.0.3		•	Macintosh System Software Greek 6.0.4
•			Macintosh System Software Hanzi (Chinese) 5.1
6.0.3		•	Macintosh System Software Hebrew 6.0.4
•	5.0	•	Macintosh System Software Icelandic 6.0.3
6.0.3		•	Macintosh System Software International 6.0.5
6.0.3		•	Macintosh System Software Italian 6.0.5
6.0.2.4	6.0.2.1	•	Macintosh System Software Kanji 6.0.4
6.0.3		•	Macintosh System Software Norwegian 6.0.5
6.0.3		•	Macintosh System Software Portuguese 6.0.4
6.0.3		•	Macintosh System Software Spanish 6.0.5
6.0.3		•	Macintosh System Software Swedish 6.0.5
6.0.3		•	Macintosh System Software Swiss French 6.0.5
6.0.3		•	Macintosh System Software Swiss German 6.0.5
•	1.2		Macintosh System Software Taiwanese 5.1
6.0.3	1.0	•	Macintosh System Software Turkish 6.0.4
Networking and Communications Products:			
•			ADSP: AppleTalk Data Stream Protocol Development Kit 1.0.2
		•	ADSP (pre-release) 1.5b4
•			Apple 3270 AP 1.1
•			A/ROSE 1.1.1
•			Asynchronous LaserWriter Driver 4.0
•			CL/1 Developer's Toolkit for the Macintosh 1.1
•			CL/1 Server for MVS/TSO 1.1
•			CL/1 Server for VM/CMS 1.1
•			CL/1 Server for VAX/VMS 1.1
•			MacAPP™ 1.1
•			Macintosh Communications Toolbox 1.0
			Communications Tools,
•			Basic Connectivity Set (pre-release) 1.0b5
			Macintosh Communications Toolbox,
•			Source Code Examples (pre-release) 1.0b5
•			MacTCP® 1.0.1
•			MacWorkStation™ 3.1
•			MacWorkStation Dialog Builder 1.0b1
•			MacWorkStation Event Handler 1.0b1
			MacX25 1.0
HyperCard® Products:			
			HyperCard 1.2.2 Update (required by System Software 6.0.3 and earlier)
•	•		HyperCard 1.2.5 Update (required by System Software 6.0.4 and later)
•	•	•	HyperCard AppleTalk Toolkit 2.5
•		•	HyperCard CD Audio Toolkit 1.0
•	•	•	HyperCard MacTCP Toolkit 1.0
2.5		•	HyperCard Serial Communications Toolkit 2.6
1.0		•	HyperCard VideoDisc Toolkit 2.0



Notes: This chart contains the latest information as of 6/1/90 and shows the latest versions of software available from three different sources: APDA™, AppleLink, and the latest Developer CD-ROM (Volume IV, named *Discy Business*). The latest version available from these sources is given after the software's name. If a cell contains a bullet (•), the source it represents has the latest version. If a cell has a number that is the version that the source has, and it is *not* the latest version. If a cell is blank, then the source it represents does not have the software. All international versions of System Software are available to Apple Partners only.

To find files on the Developer CD, Volume IV, look in the "Developer Essentials" folder at the top level of the disc.

To find files on AppleLink, look in the "Developer Services" and "Apple SW Updates" folders at the top level of AppleLink. In addition, current Macintosh software version listings (for both Apple and third-party software) are regularly updated and posted on AppleLink (pathname "3rd Party Connection: Current Versions of Macintosh Software").

We want your feedback on this new format; just link us at the AppleLink address DEV.PRESS. We want to know what you like, what you don't, and what we can do to make the information in this column more useful to you. 🍏

Sound, video, animation, and more...

CHANGING EVERYTHING WITH MEDIA INTEGRATION



We are rapidly moving toward a world in which most information and entertainment will be prepared and delivered in digital form. We are already most of the way there with text and sound. Drawings, illustrations, still images, and three-dimensional graphics

are moving in the same direction. Motion images are next. This changes everything. Everyone involved in computers, publishing, education, broadcasting, entertainment, and telecommunications will be affected. And what each of us does (or does not do) will affect all the others.

—Jonathan Seybold, industry pundit

The future is now—or at least some of it is. If you read last month's cover story in *Apple Direct* about applications of the future, you may be thinking that anything multimedia-related is just a little too far out there for you. After all, you've got an upgrade to get out, users to support, a product line to maintain. So this month, we're going to bring the subject a little closer to home and expand on a theme we brought up in that article, namely "media integration," a strategy that makes sense for a very broad base of third-party companies—today.

The term *media integration* refers to the incorporation of sound, animation, and/or video into the latest version of any kind of product: word processors, spreadsheets, CAD programs, utilities—you name it. We're not necessarily talking high-end, video-rich, laserdisc-based products. We're simply talking about the next phase in the evolution of personal-computer software. Just as

character-based products now look pretty dull compared with those that are graphics-based, a product that simply emulates a piece of paper (static and silent) will someday soon look mighty boring compared with those that add motion and sound. And the difference is more than just looks—these new applications will allow their users to work smarter and faster than they could with older, simpler applications.

Who's on First? Most major developers have projects underway that feature media integration in big and small ways. Ashton-Tate has just released Full Impact 2.0, which includes sound tools and an animated demo that “reads” information from the spreadsheet. It also includes an integrated resource tool that allows the user to add sound and picture resources easily to their spreadsheet.

Letraset has put its suite of products onto a CD-ROM disc with a HyperCard front end that features animated icons for easier navigation. Along the same lines, SuperMac has put Pixel-Paint Professional onto a CD-ROM disc that uses sound as an interface tool to help guide users. Both Microsoft and CE Software have added voice annotation to their mail products, Microsoft Mail and QuickMail, respectively (see figure 1). You can also expect to see voice annotation in the near future on several products designed for the collaborative creation of documents.

Media integration can be something as simple as animating a cursor or a dialog box—like the spinning globe and thermometer bar in AppleLink, both of which give you a sense that something's happening after you've issued a command. It can mean creating a tutorial or sales demo that walks users through your product with sound and animation. It can also mean enriching the appearance and information content of your product with these new media. Or it can mean letting users use new data types—such as animation—in their documents to express complex relationships (see figure 2,

next page.)

However you view it, though, keep in mind that *Apple is behind media integration in a big way*. At the Worldwide Developers' Conference last May, CEO John Sculley said that Apple is moving a number of multimedia technologies from research to product status. At the WWDC, Apple also announced QuickTime, an extension to the Macintosh Operating System that will be analogous to QuickDraw (for more details, see “The Lowdown on QuickTime” on page 13 of last month's *Apple Direct*). The idea behind QuickTime is to provide a cross-platform standard for the control of media devices and multiple data streams.

Why Start Now? Apple plans to release the QuickTime specification by the end of this calendar year. So why should you start investigating media integration now? Well, for starters, by getting involved immediately, you have an opportunity to *influence* QuickTime. Apple is calling for developers to submit their ideas for the QuickTime architecture, so the more savvy you are about the issues and how they pertain to your development goals, the more input you'll have into the process. You can join the dialog by posting your comments on AppleLink, in the folder whose path is “Developer Services Bulletin Board: Developers Ask Each Other: Media Integration Discussion”.

More to the point, though, is that by developing with currently available tools, you'll be

By Gregg Williams
& Lisa Raleigh,
Apple Direct Staff

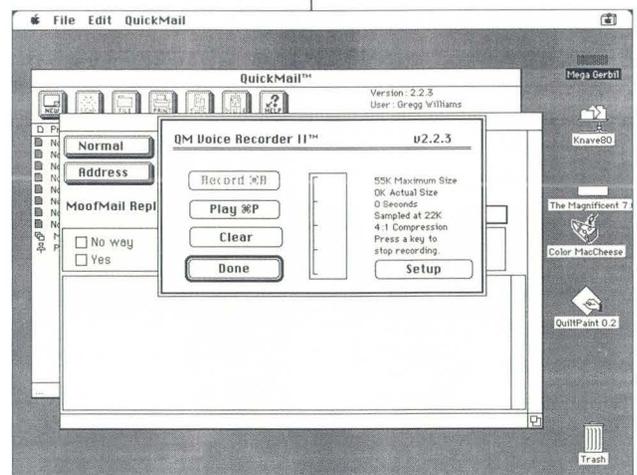


FIGURE 1:
Voice annotation in an existing application. As this example (CE Software's QuickMail, using Farallon MacRecorder software) shows, sound input can be made very easy to use.

better prepared to take full advantage of QuickTime and other tools when they become available. (See Table 1 on page 11 for a summary of present and future tools.)

The learning curve for media integration promises to be challenging, because it involves communicating with users in new ways. Evangelist Scott Mize, for instance, recommends hiring people skilled in particular media forms, such as audio or video producers, as part of your development team when these new media are



FIGURE 2:
Animation helps users grasp dynamic processes. Not only can it show the dynamic data, but it also helps users look at and analyze different views of the same data. Here, a student uses the data from the bouncing ball simulated on-screen to study the physics involved.

involved. (This goes along with Evangelist Bruce Tognazzini's advice that you hire a designer when creating a program's user interface.)

Another benefit of working with new media now is that you'll also be in a better position to strike a deal with *content providers*, if

it fits with your business goals to do so. Media integration opens the door to new types of information in the personal-computing environment. More and more, Apple is seeing (and encouraging) partnerships between software developers and the owners of information, such as book publishers and television producers. The developers who have the best handle on media integration will also have the most to offer an information provider looking for a partner to create multimedia products. As one example of this, see the "FYI" story on Dow Jones' Alliance Developer Program, on page 5.

In the rest of this article, we'll look at some of the tools that are available today (or that will be, in the near future) to help you prepare for the day when media integration is as common as bit-mapped graphics. (If you have or know of a tool that's not described here—and you want other developers to know about it—you can post pertinent information in the Media Integration folder on AppleLink (using the pathname given above).

"Print to Videotape." Don't you wish you could do that—finish a presentation, select Print to Videotape from the File menu, and end up with

a videotape of your presentation that you can play in any VCR? Well, you can, and here's how: Install any of the new Apple Display Cards (4•8, 8•24, or 8•24 GC), attach an encoder box to it, hook the output to your VCR, and start taping! *You should start thinking of the VCR as an inexpensive color-output device for the Macintosh.*

The Display Cards were designed with VCR taping in mind. They provide timing support for NTSC (U.S.) as well as PAL (foreign) broadcast television standards, so you need only an encoder box to convert a Display Card's output to a form that can be videotaped. (Encoder boxes cost from \$250 to several thousand dollars, depending on your needs. For example, one encoder box in the mid-price range is the RasterOps Video Expander, which sells for \$395.)

Does it make sense for your customers to videotape the video/audio output of your program? In many cases, the answer is an overwhelming *yes*—desktop-presentation programs, for example. With other programs—word processors, spreadsheet programs, CAD packages, or whatever—your customers may benefit from videotaping your program's output, if only to create tutorial and training videotapes.

You can give your customers the print-to-videotape capability with very little effort—just tell them to use an Apple Display Card and an encoder box, and give them some ideas about what they might want to videotape. Still, there are a few factors that may necessitate slight changes to your program:

- *Mind your colors.* If you're using 8-bit color, make sure the colors in your video palette look good on videotape. Some colors are outside the range that broadcast video can reproduce, so check the colors on an NTSC (in the U.S.) or PAL (in Europe) monitor before you do any significant work.

- *Don't lose information off the edges.* A television ensures that the video image fills the screen by doing what is called *overscan*—enlarging the video image so much that some of the contents fall off all four edges of the picture. Consequently, you may want to have NTSC and PAL "page" sizes built into your printer-setup dialog box, to ensure that all of the computer's image will be visible on a television or monitor. The applicable rule of thumb says that your data should fall in the center 85 percent of the video display.

Table 1: What You Can Do with Media on the Macintosh

To do this:

...you'll need this:

Now

print to videotape	modular Macintosh, Apple Display 4•8, 8•24 or 8•24 gc video card, encoder box, VCR
create a "tape" simulating any activity on the Macintosh screen, annotated with graphics and sound	MediaTracks (from Farallon, commercially available)
create interactive tutorial//training materials	Apple's HyperCard 2.0 or Silicon Beach's SuperCard
manipulate animated, speaking humanlike face images	InterFACE (from Bright Star Technology); usable from several major software programs; contact Bright Star for licensing code that will allow your program to control face images
control (from your program) arbitrary animations in color or monochrome, with sound	MacroMind Multimedia Manager (contact MacroMind for licensing details)
control audio CD and CD-ROM players, videodiscs, VCRs, and other media devices	HyperCard CD Audio, Serial Communications, and VideoDisc Toolkits (all available from APDA)

By end of 1990 (System 7.0)

do voice annotation or sound recording from inside your program	Articulate System's Voice Link or Farallon's MacRecorder system (both require licensing code from the product's originator)
play multiple channels of sound simultaneously	System 7.0 Sound Output Manager, from Apple
play sampled sound direct from hard disk	System 7.0 Sound Output Manager, from Apple
do sound compression and expansion	System 7.0 Sound Output Manager, from Apple
do easy or sophisticated voice input	System 7.0 Sound Input Manager, from Apple

Later

take control of multiple audio and video recordings	QuickTime (to be furnished by Apple with a future release of system software; specification to be finished by the end of this year)
convert text (or phonemes) to computer-generated speech	to be delivered by Apple by May 1991
do software-only video compression/expansion	to be delivered by Apple by May 1991

- *Plan for output to a monitor or television.* There are a lot of small details that you can take care of so your customers don't have to. Don't do any light/dark single horizontal lines, that are an odd number of pixels high; these may cause a flickering image (even when using Apple Convolution, which minimizes this problem). Make sure your fonts are large enough to read and that the colors you use work well together. Avoid very bright colors (such as bright red), because they may shimmer along the edges when displayed.

- *Avoid large blocks of type.* Video is good

for presentations (a few words at large point sizes) but not for extended reading (multiple lines of text at smaller point sizes).

Tutorial and Training Tools. If you want to provide on-line tutorials, training tools, or marketing support materials with your product, you have several useful products from which to choose. Farallon Computing's MediaTracks is good for showing how a piece of software is used, while Apple's powerful new HyperCard 2.0 and Silicon Beach's SuperCard give you two capable ways of creating more general tutorials.

Sound
annotation
will probably
be the first
new-media
addition you
make to your
program's
documents.

MediaTracks essentially marries and enhances two of Farallon's best products: MacRecorder, for recording and playing back sound, and Screen Recorder, for recording and playing back whatever is happening on the video display. With MediaTracks, you can record a monochrome video sequence, then edit it, enhance it by drawing in color "on top of" the video, and tie recorded sounds to any position in the video "tape." You can also play a tape from within HyperCard, and the tape can interact with HyperCard via callbacks. (MediaTracks tapes can be played from a HyperCard stack or a supplied "player" application that is freely distributable; in addition, MediaTracks can generate a double-clickable file that contains both the tape and the code needed to run it.)

HyperCard 2.0 and SuperCard are good for many kinds of projects. We are interested here in their ability to create interactive *stacks* (for HyperCard) or *projects* (for SuperCard) that display text, graphics, and simple animation driven by user mouse and keyboard input. HyperCard 2.0 has many new useful features (see "HyperCard 2.0: a Good Tool Gets Better," by Gregg Williams, in the May, 1990 issue of *Apple Direct*); it's good to use because it comes standard with every Macintosh. SuperCard, on the other hand, can do some things HyperCard 2.0 can't (like more arbitrary use of color), and you can "compile" a SuperCard project into a stand-alone application that you can distribute freely.

Agents: Use with Caution. You've probably heard the story about the dog that walked on two legs—the point was not that it walked poorly but that it walked at all. We now have computers with enough horsepower to display, well, *talking heads* in our programs. The question is: When should we?

An *agent* is more than just a talking head—it is a semi-intelligent, semi-independent software entity that can handle routine tasks, without your supervision, while you are doing other things. When a software entity takes on that aura of having a "life" of its own, a human-like visage seems helpful—but you should make sure that your use of talking-head technology is appropriate and not just a matter of "Gee whiz, look at that." You should look at both your subject material and the intended audience when deciding whether or not to use a humanlike agent.

If you are interested in offering an on-screen talking head, Bright Star Technology, which has had its HyperCard-based HyperAnimator product out for some time, now has a more general and powerful product called InterFACE, which allows you to create 8-bit-color animated faces (which the company calls *actors*) of various sizes and lets you use them from HyperCard, SuperCard, MediaTracks, and a handful of other applications.

More importantly for you as a developer, Bright Star has announced that it will license its actor technology for a flat fee of \$500 per year; it will supply you with the code you need to control the actors' face and voice from within your program. For more information, contact Bright Star Technology at (206) 451-3697 or AppleLink BRIGHT.STAR.

Animation Helper. If you want to do animation on the Macintosh, you might be interested to know that MacroMind is licensing a product for developers called the MacroMind Multimedia Manager (M5), which is a set of resources and glue routines that lets your THINK C or MPW Pascal program play a MacroMind Director scene in a separate window or on a full screen. (It can run MacroMind Director 1.0 and 1.0.1 files, VideoWorks files, and MacroMind Director 2.0 files that aren't interactive and don't use the Lingo scripting language.) MacroMind requests that you contact it about M5 via the AppleLink address M5.

Coming Soon: Integrated Sound I/O under System 7.0. The items above are available for use today, but there are several important sound-related technologies (see table 1) that are waiting for System 7.0, which will be in your hands Real Soon Now and in users' hands sometime before the end of this year.

Because sound will probably be the next medium you'll add to your programs, Apple wants to do an especially good job of integrating it into the everyday Macintosh environment, and this means starting with a very solid foundation and building from there. Because of this, *Apple recommends that you not talk directly to any sound hardware but instead build on top of the new Sound Manager* (which will be available soon).

Articulate Systems, Inc., has developed a technology that allows third-party developers to control two Articulate Systems products—its

Voice Navigator and Voice Link—from the developer's program. (The Voice Navigator is Articulate Systems' Macintosh voice-input product, and Voice Link is its high-speed audio digitizer, which has audio filtering and compression/decompression in a DSP—or digital-signal processor—chip.) Articulate Systems will be licensing its software, which is also built on top of the Sound Input Manager, to selected developers. For more details, contact Dana Morgan of Articulate Systems at (619) 876-5236 or AppleLink ASI.SUPPORT.

Farallon Computing has created a lower-cost version of its MacRecorder hardware, called the MacRecorder Voice Digitizer, meant to be bought in quantity and distributed to all the Macintosh users who work together so that they can send and receive voice messages. Farallon is interested in helping developers incorporate sound recording and playback in their products; their technology also builds on top of the Sound Input Manager. For more information, contact Farallon at (415) 596-9000 or AppleLink FARALLON.

With the new Sound Input Manager and an improved Sound Manager, Apple hopes to provide a hardware-independent sound I/O platform that programs can use on all Macintosh computers. See the "Making Macintosh Sound More Powerful" sidebar for more details on the new sound-related features.

Sound Bytes for Your Program. We can't say it too clearly: *Sound annotation will probably be the first new-media addition you will make to your program's documents.* Whether your program is a desktop-publishing program, a MIDI sequencer, or an outline processor, users will probably find some use for attaching a voice message to some part of a document (a word, object, or field, for example). As we mentioned earlier, the two most popular electronic-mail systems, QuickMail and Microsoft Mail, already support voice messages via the MacRecorder Voice Digitizer, and several other (nonmail) programs that let users do voice annotation are already in the pipeline.

Imagine how useful sound annotation might be. An art teacher might scan in reproductions of famous paintings, annotate them with voice messages, and leave them on a file server for students to study at their leisure. A literacy volunteer might quickly put together a HyperCard stack that would help a client concentrate on the spelling and sound of difficult

Making Macintosh Sound More Powerful

Apple has known for quite some time that sound would be the next medium to be added, so our engineers have been working on enhancing Macintosh sound at the Toolbox level. The new version of the Sound Manager adds major new features to the manager, and the new Sound Input Manager gives the other half of the sound "coin" to all Macintosh computers. The engineers have come up with a system that maximizes (with trade-offs) the use of sound in several areas that are important to Apple: power of new features, independence from hardware configuration, applicability to the entire Macintosh product line, and compatibility with existing software. Apple strongly recommends that you do all your future sound-related development on top of this foundation.

- *Blending multiple channels of sound.* One of the most needed enhancements to Macintosh sound has been the ability to play multiple channels of sound simultaneously. Under the System 7.0 Sound Output Manager, most of the Macintosh product line will be able to blend multiple channels of sampled sound from one application or combine single channels of sampled sound from multiple applications. As you would expect, the more powerful Macintosh models can blend more channels of sound and produce better-quality sound than the less powerful ones. Only two models, the Macintosh Plus and the SE, lack the resources needed to blend multiple sound channels while leaving enough computing resources for other tasks. *Your program should use the Gestalt Manager to determine whether or not the computer on which it is executing has the features you need.*

- *Continuous play from disk.* Another interesting capability of Macintosh models with the Apple Sound Chip is its ability to play sampled sound continuously from a SCSI device (hard-disk or CD-ROM drive, for example) while simultaneously executing other tasks. You can also use your Macintosh to play background sound while you work, but we'll leave it to talented developers such as yourself to come up with some additional creative uses for this feature.

- *Sound compression and expansion.* The MACE (Macintosh Audio Compression and Expansion) routines, which are available through APDA for use with existing Macintosh system software, have officially been added to the System 7.0 Sound Output Manager; these routines work with all Macintosh models, starting with the Macintosh Plus. Again, the capabilities of these routines vary with the power of the Macintosh computer, but all the supported models can do 3:1 and 6:1 audio compression (the 68020- and 68030-based models can also do it in real time) and real-time 3:1 and 6:1 expansion and playback of sound.

Apple is making voice input a reality: All future modular Macintosh models will include sound-I/O capabilities, and existing Macintosh computers can easily be retrofitted to provide them as well.

The Sound Input Manager is a device-independent architecture that will allow programs to do sound input without knowing the specifics of whatever sound-input hardware (including third-party digitizers) is present. Its new function calls allow for the simultaneous recording of multiple channels of sound and simultaneous recording from multiple input devices. Other features include high-level (simple, easy to use) and low-level (detailed, high level of control) sound-input calls, stereo recording, multiple sound-quality rates, and recording directly to hard disk.

words. An engineer and a machinist might attach voice messages to a gear object in a CAD document and send it electronically to their boss instead of waiting for a face-to-face meeting. A salesperson might use it to creatively explain certain questionable items on his expense account (although technology can help only so far).

It goes without saying [sic] that voice is one

of the most important forms of communication we have—if you don't think so, try to go for an entire day without talking. So far, computer documents have lived under a vow of silence. Imagine how much more useful they will be

when they can combine voice, text, graphics, and animation.

(One thing curiously missing from this article is any mention of human-interface issues. *How do you show that a spreadsheet cell has voice in it? What should happen when you print a document with voice—or even video—in it?*

Apple is working on that and will deliver to you, sometime this summer, human-interface guidelines for new media. You should visit the "Apple Human Interface" discussion in the "Developer Services" bulletin board of AppleLink; you can ask questions there and get answers from both Apple and third-party people. As a rule of thumb when you can't decide which interface choice to use, make several choices available and allow the user to make one a default preference.

On the Horizon. Given Apple's deep commitment to integrated media, it should not surprise you that Apple's plans extend beyond the items we've already discussed. We want to say something about these plans so that you can plan your product strategy. These innovations—text-to-speech; video compression/decompression system software; and, later, QuickTime—will probably start occurring in the 1991 time frame.

Voice Output on a Budget. One good way to involve your users with your product is to give

them spoken output to listen to, but digitized speech output does not come cheaply. Voice files can easily grow beyond a megabyte of storage space—which is definitely a problem with floppy-disk-based storage and is sometimes a problem even on hard disks (but, thankfully, is *not* a problem on CD-ROM discs—yet). In those cases in which the identity of the speaker is not important (on-line documentation, for example), a high-quality text-to-speech algorithm can deliver minutes or hours of speech for less than a kilobyte of disk space per minute.

Apple's ATG (Advanced Technology Group) is currently investigating a new form of text-to-speech technology. (Apple expects to deliver text-to-speech software in the first half of 1991.) One ATG investigation is a software-only text-to-speech solution for the Macintosh. It is based on natural human (not synthesized) speech and works with one or more "plug and play" voices (e.g., adult female voice or male child voice). The technology accepts any combination of ASCII text and phoneme strings and can be localized in the future for non-English languages.

Apple's goal here is to create computer-generated speech that sounds natural and is easy to listen to. Current text-to-speech algorithms (including MacinTalk™) fall very short of that goal, which is why computer-generated speech has not gained widespread use in the personal-computer community. Once a good text-to-speech technology is available, you will be able to create programs that use speech as freely as they now do text or graphics (see figure 3). This, again, will result in Macintosh products that offer significant advantages over those on other platforms.

Video on a (Meatier) Budget. If you think sound files get big quickly, you may not want to know how big video files get. An 8-bit-color image in a small 250-by-300-pixel window takes about 75K of memory for *one frame*. Multiply that by the rate of 30 frames per second, and you need about 2.25 *megabytes* of storage for one second of uncompressed video!

Video compression/expansion will be one of the hottest and most important technologies of the 1990s. Hardware will, of course, provide the fastest and most efficient solutions—but not everybody will have the additional hardware

Continued on page 24



FIGURE 3:

Voice output will allow the Macintosh and the user to do things neither could do before. Voice output allows the Macintosh to communicate a large volume of data or give feedback when the user's hands or eyes are directed elsewhere. Here a spreadsheet helps the user proofread a set of numbers.

APPLE SURVEYS ITS CUSTOMERS

A significant proportion of them are 7.0-capable

In the June issue of *Apple Direct*, we shared some data from IntelliQuest that indicates what types of products Macintosh users are planning to buy. But let's drop back a bit.

What is it that Macintosh customers are already using?

Knowing what the installed base of Macintosh configurations looks like is vital to your business. How many megs do users typically have? What types of peripherals do they use? How often do they buy new applications?

Apple recently completed a research study to answer these questions (and many more). The survey involved almost 2,000 Macintosh users from 5 countries (the U.S., Canada, Australia, France, and Germany), with about half of them from the U.S. The study explored the types of configurations these users had and their general satisfaction level with Macintosh.

Representative Sample. The survey respondents all acquired their systems between 1986 and 1989, and had one of the following systems: the Macintosh Plus, SE, SE/30, Macintosh II, IIfx, or IIfx. The sample represented all the markets in which Apple participates—business, education, home, and government. About a quarter of them have four or more years of experience on the Macintosh (meaning *all* Macintosh experience and not just the system they're currently using), while 38 percent have between two and four years, and the remainder less than two.

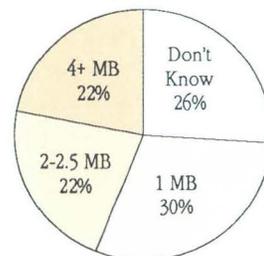
For reasons of confidentiality, we can't share all the results with you, but we can give you some of the highlights. In general, these users seemed to possess a fairly high degree of sophistication. For instance, about two-thirds of them knew the version of the system software

they were running.

The study also found that a very high percentage of Macintosh users have at least 2 megs of memory (and are thus already System 7.0-capable), that hard-disk usage is quite pervasive, that scanners and modems are widely-used peripherals, and that Macintosh customers tend to use many applications and continue to buy more.

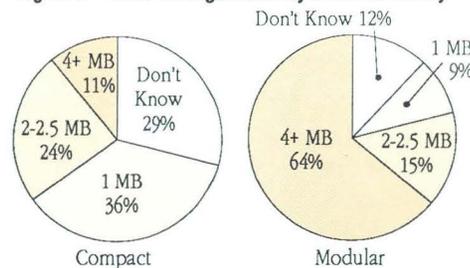
Breaking the 2 Meg Limit. Close to half the respondents to the survey had at least two megabytes of memory in their systems. (Fig. 1.)

Figure 1—RAM Configurations



The picture shifts a bit when we break down the data between compact and modular users. Even then, more than a third of the compact users have 2 megabytes or more, while the figure goes

Figure 2—RAM Configuration by Product Family



up to almost 80 percent for the modular users. (Fig. 2)

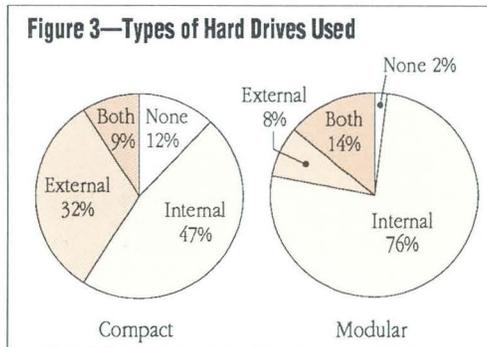
By Lisa Raleigh,
Apple Direct staff

These Mac users seemed to possess a high degree of sophistication.

Almost 50 percent said they had purchased an application within the last three months.

Hard Drive Heaven. Hard-disk usage is 90 percent overall; when 1989 purchasers are considered separately, the overall figure goes up to 94 percent.

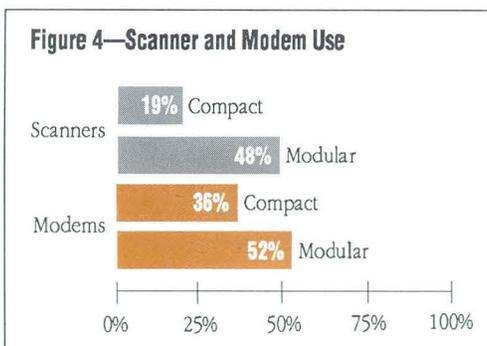
Interestingly, a fairly high number of users have *both* internal and external hard drives



(Fig. 3). Among compact users, nine percent have both, while 14 percent of the modular users do.

As with internal memory, the pervasiveness of hard-drive configurations means that a significant proportion of the Macintosh installed base is already prepared, hardware-wise, for System 7.0.

Scanners for the Rest of Us. What percentage of Macintosh customers would you guess are using scanners? If you said 25 percent, you're in tune with the respondents to this survey. This figure



goes up to 48 percent when just modular users are considered (Fig. 4).

The study didn't inquire about scanner *ownership*, but scanner use instead, so it's likely that many of these users are sharing a scanner with others. Nonetheless, what implications does this high rate of scanner use have for your products? How might your products, documentation, or tutorials be modified in order to take advantage of it?

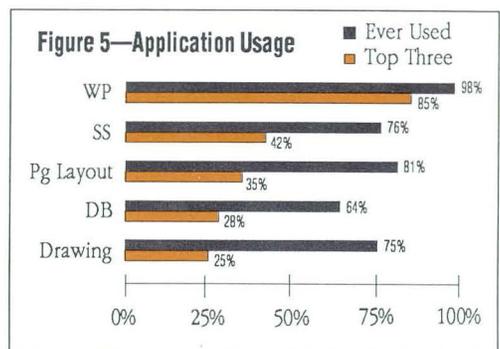
Along the same lines, these users show a high level of modem usage. Overall, almost 40

percent of the respondents said they use modems, and that figure climbs to 52 percent for modular users.

Lots of Applications. A couple of years ago, Diagnostic Research performed a survey called "Macintosh or MS-DOS?" that showed that Mac users tended to use more applications than DOS users. While this more recent survey didn't make the same comparison, it did find that the respondents generally use many applications.

Overall, the median number of programs for compact users is six, and the figure goes up to nine for modular users. And good news for developers—users of both types of systems continue to buy more software. Almost half of the survey respondents said they had purchased an application within the previous three months, and more than 80 percent said they had purchased one within the previous year.

Not too surprisingly, word processing is the most popular "ever-used" application, with almost 100 percent of the respondents saying that they had used it at some time or another. It's also the most popular application when users are asked to name their "top three" types of



programs (Fig. 5). However, there's a great deal of diversity among other applications also given "top three" status, strengthening the observation that Macintosh users tend to use their systems for a broad range of tasks. (See the IntelliQuest MediaTrack results in the June issue of *Apple Direct*, which indicate a similar trend.)

All's Well That Ends Well. How did these Macintosh users feel about the Macintosh overall? Very favorably. More than 90 percent of them gave the Macintosh a "5" or a "6" on a six-point satisfaction scale, and three-quarters of them said they were "very likely" or "extremely likely" to recommend the Macintosh model they use to others. 🍏

By Jessa Vartanian and Scott Converse

System 7.0, Tools, XCMDs, and More

In this, the second appearance of our new column, Electronic Media, we look at several information sources that we think are critical to your continuing success. As you read what follows, you'll notice that many of the items we've chosen to discuss stress interactivity (is that a word?) That's 'cause we want to hear from you (and we know you have a thing or two to tell us). You'll also find mention of numerous tools we know will make your lives a little easier. And then there's that new system software.... So go ahead and have a look, and when you get a moment, let us know what you think: AppleLink: DEV.PRESS.

System 7.0. Now that you have the *System Software 7.0 Alpha Development Release CD* in your hands, you'll want to know about any and all updates to it. If you regularly check the System Software 7.0 folder on the Developer Services Bulletin Board (DSBB), you won't miss a beat. Documentation changes and additions (e.g., AppleEvents Registry updates and Tips for Using Inside Mac Volume VI Online Edition) and any new technical notes, sample code, and the like will be distributed on line as they become available. (AppleLink path: Developer Services: Macintosh Developer Technical Support: System Software 7.0.)

And while you're getting to know our new System software, participate in the System 7.0 discussion in the DSBB System Software 7.0 Discussion folder. (AppleLink path: Developer Services: Developers Ask Each Other: System Software 7.0 Discussion). For easier access to this folder (or any other folder, for that matter), you can program the System Software 7.0 and System Software 7.0 Discussion folders into your AppleLink Personal Menu. (Note that these two System 7.0-related folders on the DSBB are restricted to Apple Partners.)

CTB: Not Just Another Acronym. The Macintosh Communications Toolbox, or CTB, is gaining ground among developers, so we've devoted a folder to it on the DSBB. In the Macintosh Communications Toolbox folder, you'll find CTB information from Apple—data sheets on the Communications Toolbox and the Basic Connectivity Set, Q&As, listings of third-party products that support the Communications Toolbox, what have you—as well as general information on new developments in the Communications Toolbox world.

But enough from us; the Communications Toolbox folder wants to hear from you too. We've set up areas within the folder to which you can contribute your own tools and sample code for other developers to experiment with and download. We've also set up a general discussion folder for those of you who want to swap questions or suggestions or discuss problems you're having with the Communications Toolbox. Apple's Communications Toolbox team has been rumored to put in their two cents worth of discussion here too, so don't miss the opportunity to communicate with them directly. (AppleLink path: Developer Services: Macintosh Communications Toolbox.)

For those of you who want to write Communications Toolbox tools, we've provided some example skeleton tools on *A Disc Called Wanda*. You'll find examples for each of the following Communications Toolbox managers: Connection Tool, Terminal Tool, and File Transfer Tool. In addition, don't miss Surfer v1.0.1d1—a Pascal-based shell application that uses the Communications Toolbox. This source code enables developers to see how calls are actually made in a CTB application (Developer CD Series path: A Disc Called Wanda: Programming & Utilities: Moof: Comm Toolbox Source Ex.)

AppleLink on AppleLink. You can now sign up for additional AppleLink accounts via AppleLink itself. If you already have a Developer AppleLink account, you have access to the on-line Developer AppleLink Account Application. Just download the application

Continued on next page

Scott Converse (CONVERSE1) is Electronic Media Group Manager in Apple's Developer Press Group. Jessa Vartanian (JESSA) is Developer AppleLink Program Manager in our Developer Programs Group.



*The Support
Tools eXternals
stack on
Wanda con-
tains more
than 40 useful
XCMDs for
you to play
around with.*

Electronic Media

Continued from previous page

(after you've read the AppleLink Overview and Pricing Addendum information posted alongside it), fill it out, and Link it back to us (unless you are using a purchase order, in which case you will have to return it to us via U.S. mail). We'll start processing your account requests right away. (AppleLink path: Developer Services: Developer Programs: Developer AppleLink Program.)

Wanda's Taking APDA Orders. Did you know that there's a stack version of the spring 1990 *APDAlog*™ on *A Disc Called Wanda*? What's more, you can use the stack to create an APDA order and then send that order to APDA via AppleLink. Here's how it works. First, you browse through the stack. When you find what you want, click the product's part number. The stack will ask you whether you want to add that product to your order, and if you do, it'll ask you how many units you want. Pretty smart, huh? Respond to the prompts (this will automatically add the product to your order card), and then go on to the next product. When you're done ordering, click the Order Card button, fill out all the pertinent information, and then either click the AppleLink button to have your order placed directly in your AppleLink Out Basket (you'll need to supply a credit-card number to play this part of the game), or print it out and mail it to APDA. (Developer CD Series path: A Disc Called Wanda: Info Island/SpInside: APDA.)

Remember, it's also a good idea to check AppleLink for updates to the quarterly *APDAlog*. As soon as new products are available through APDA, their description, part number, and price are posted on the DSBB. (AppleLink path: Developer Services: APDA: Updates to Quarterly APDAlogs.)

Registration for DU Courses. The Developer CD has also made Developer University more accessible. *A Disc Called Wanda* contains a HyperCard stack with which you can peruse DU's schedule of classes, select those you want to take, find out where they are and how much they cost, fill out a form conveying all this, and then send that form off to Developer University via AppleLink. How'd we do it? We use Mitem technology (from Mitem Corporation in San Jose) to create a link between HyperCard and an AppleLink address. You just choose and click, and *shazam!*, you're registered. Of course, the part where you send in a check is still required. DU will Link back the particulars. (Developer CD Series path: A Disc Called Wanda: Info Island/SpInside: Developer University.)

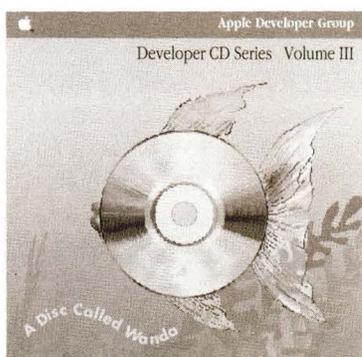
develop. As you may already know, *develop*, *The Apple Technical*

Journal, is now on *A Disc Called Wanda*. When you received your *develop* magazine, you may have noticed that the CD-ROM disc that was in the first issue was missing in the current issue. Well, it wasn't an oversight. *The Developer CD Series* CD-ROM disc goes out on the same schedule as *develop* (and in the same monthly mailing), so we decided to put it all together on one disc for you. This way you don't have to dismount *Wanda* to get at the electronic version of *develop*. If you have subscribed to *develop*, you'll still get the CD-ROM disc, bound into the magazine, that has the current and all past issues of *develop* right there, on line and accessible. Don't forget, we've included on the disc all the sample code that each article talks about (the sample code in each issue of *develop* is also cumulative) so you can just cut and paste. No more tedious retyping of code (jeez, we hate that—don't you?).

Macaroni. Macaroni is an interactive training tool for helping people learn the Macintosh Toolbox. It gives curriculum developers a medium for writing interactive courseware about Macintosh programming. Macaroni, in essence, visualizes, in a simple and direct manner, the most popular calls from Apple's classic reference *Inside Macintosh*. Calls are presented as fill-in-the-blank forms in an exploratory environment that's fun to use. The forms can be scripted and visualized during execution. Cool stuff. Adam Bartlett, Rich Millet, and Jonathan Zar are the authors of this stack, and many an aspiring Toolbox whiz will appreciate the work they put in to create this great HyperCard software-based training tool. We suggest you use it in conjunction with *SpInside Macintosh* to really get wired into the Macintosh environment quickly, or to brush up on long forgotten calls (no, not the ones you should be making to your mother; we're talking a totally different Toolbox here, folks).

Late-Breaking News. All the news that fits, we print. As would be appropriate with any ol' quarter-million-page electronic magazine, we've added a place on *Wanda* to put the last-minute stuff—the hot-off-the-debugger code. One of the items in this juicy little folder is called Support Tools eXternals. It is brought to you by those two wild and crazy guys Eric "CYNIC" Carlson and Anup "SKEPTIC" Mararka (those middle names happen to be their AppleLink addresses, but don't tell them I told you). This stack has 44 extremely useful XCMDs that you can play around with. We used many of these XCMDs in the navigational stack that came on *Wanda*, and we thought you might find them as useful as we did. Here's a partial list of the XCMDs Carlson and Mararka have included in the stack: ApplMemory, ApplSize, AUXisRunning, CDDriveList, CopyStruct, DiskIsRemovable, Dis, SpaceAvail, DocumentCreator, DragRect, EjectDisk, FileCreationDate, FileCreator, FileExists, FileModDate, FilePath, FileSize, FileType, FileVersion, FolderContents, FolderPath, HFSCopy, HFSRename, INITList, MenuShowing, MFIsRunning, MonitorCount, MonitorConfig, NumToBinary, NumToHex, OpenFiles, PlotFinderICN, PrintContainer, SCSIInfo, SCS.

This is obviously a stack not to be missed. (Developer CD Series path: A Disc Called Wanda: Late-Breaking News: Support Tools eXternals.) 🍏



By Bruce "Tog" Tognazzini

Intuitive & Consistent?

Dear Tog: I recently started working for a Macintosh programming firm. My previous programming experience was with minicomputers, but I had always been attracted to the Macintosh, with its snappy-looking displays, quick interaction, and consistent user interface. Even some mainframe programmers, often derisive toward personal computers because of their small MIPS ratings and "toy" operating systems, were impressed by the Macintosh's versatile user environment and interapplication cut and paste. No career programmers, in my experience, cared too much about its being "intuitive" or "consistent." This is fortunate, because after using a Macintosh for more than two years, I have found it to be neither intuitive nor consistent.

For example, examine the Macintosh use of icons. I agree that they are more versatile and interactive than a typed filename or device name, and a lot harder to misspell. But I don't believe they are somehow more "intuitive," especially when the icon bears no resemblance to the visual appearance of the device or document it represents. Noting that applications have a little hand in their icon is no more obvious than noting that their names end in ".EXE" and is harder to enforce. More often, icons are used to demonstrate their creator's artistic whimsy or as free advertising space in which to display the company logo as prominently as possible.

And what of "consistent"? You say clicking an icon selects it? Not if it's in the menu bar; or in a program in which it activates a command; or in a dialog box in which it may be selected, activate a command, create a menu, switch you to a different task, or do nothing—with or without being highlighted first to warn you that something may or may not happen. In fact, creative new icon subspecies (such as the highlighting nonselectable option-toggle menu-bar icon) are appearing on my desktop constantly.

My example was of icons, but there are many others. I have used applications in which the horizontal scroll bar changes grouping in a database and menus disappear or change contents, and I've also run across word processors in which Command-P prints the document (Command-T changes the font style to Plain). There are cases where interapplication cut and paste, the most envied Mac feature in the minicomputer world, is not supported by high-end programs.

So what can anyone do about it? I can try to be conscientious,

and you can carry the gospel to Mac developers; but in the end it's the users who say what interface gets accepted.

—Richard Hough, software developer, Vancouver, B.C.

Tog Responds: The Macintosh computer sports the most jumbled-up, inconsistent, confusing interface ever made, with the exception of those of all the other computers.

I spend a lot of time in this column talking about consistency, just because it's so hard to achieve and maintain. The kinds of degradation of the original interface that Richard is addressing are quite natural: Unless we constantly work to counterbalance the effects of these sorts of "noise" creeping into the interface, they will continue to arise.

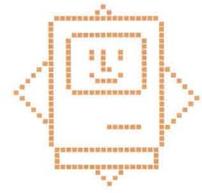
(*Noise*, by the way, is a formal term, used in information theory, that identifies any kind of data that obscures the desired message. Information theory will be the subject of an upcoming column.)

I don't spend a lot of time in this column talking about intuitive interfaces, because there ain't no such animal. Just what does *intuitive interface* mean? Quite literally, for an interface to be intuitive, it must be able to intuit—to perceive the patterns of the user's behavior and draw inferences. No commercially available interface with which I am familiar is capable of this rather daunting task, although some might lay claim to the most rudimentary capabilities.

Perhaps the engineer who first coined the term really meant "intuitable," that users could intuit the workings of an application

Continued on next page

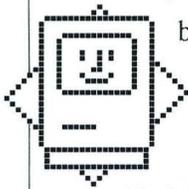
.....
This column is your input into the Apple system. Send Tog questions, comments, tirades, and editorials on the state of Apple's human interface. Include your name, city, and state—and company name, if you wish. He does not have the time to respond to each letter personally, but he will print those that strike a nerve and pass the others on to Apple Engineering. Rest assured, your words will not go unnoticed.



*It's up to
 each of us
 to keep the
 interface
 together.*

The Human Interface

Continued from previous page



by seeing it and using it. But even that is a less than useful goal, since only 20 percent of the population depends on intuition to understand anything (see my column in *Apple Direct's* October 1989 issue).

The Articulate Interface. I propose that we replace *intuitive* with *articulate*. First of all, *articulate* fits rather well with what I see as the job of the interface: speaking clearly to users through words meaningfully arranged. In the case of the Macintosh, those words are often objects—such as icons and control structures such as close boxes, scroll bars, and pop-up menus—but they are words nonetheless.

Second of all, this word *articulate* has been dying for work, having fallen into general disuse—since the passing of Winston Churchill and the arrival of the “sound bite”—as an adjective applied to people. (You can now make a two-hour campaign speech, secure in the knowledge that your media specialists will extract for the TV ads the only 15 seconds in which you didn't sound like a total boob.)

Human-Interface Objects. As I said in my November 1989 column, the Apple Human Interface is made up of defined objects. For example, there is the file-folder object, which consists of not only a standard visual appearance but also of standard behaviors: opening with a double-click, displaying contents in the user-selected view, and so on. The appearance and the behaviors are permanently bound together: Neither the programmer nor the user can (easily) create a file folder that acts like a trash can, an application, a scroll bar, or any other standard object in the interface. This binding together was intended to ensure a predictable environment for end users.

What has happened is that we within Apple and you developers out there have been letting a lot of noise get into the system: We've developed a more complex Control Panel with icons running down the left side. Did we require double-clicking? No. Are people confused? Yes.

We've popped a small icon of the current application into the upper right corner of the menu bar. Does holding on it cause a menu to fall down? No. Instead, people are switched to the application they're least likely to desire. Are people confused? Yes.

Developers have invented weird objects such as movable modal dialog boxes that look like modeless dialog boxes, document icons that look like application icons, and the Undo menu item that intuitively when the user might need it and instantly turns gray. (Maybe there is such a thing as an intuitive interface.)

Rules for Maintaining Continuity: It's up to each of us to keep this interface together. Here are a few things we can do that will help alleviate the problems Richard has found:

- Use icons in the menu bar only as menu titles. We are

hoping that System 7.0 will remove our own violations of this canon. We are hoping that our developers will do their part too.

- **Make all icons require double-clicking.** We at Apple have been particularly remiss about this one. Beginning with the iconic buttons in Find File, we have moved on to such violations as the column of icons in the Control Panel for deciding which cdev you want to access. I am happy to report that we are correcting at least most of our own violations in System 7.0. Most specifically, we will now require a double-click on Control Panel icons.

- **Make any iconic-looking object that requires a single-click also accept a double-click.** For example, in the Control Panel, the user sets a new background scheme by single-clicking on the little image of the screen. Currently, double-clicking on the screen fails to set the new background. This leads people to form very bizarre rules for setting a new background, including the rule of one user I talked to who believed that setting a new background required three clicks. Generally it's a good idea to assume that people will double-click everything, once the mood is upon them, so plan for the least surprising and least destructive result. (I've decided to leave HyperCard out of this discussion. For my feelings about the HyperCard interface, see my November 1989 *Apple Direct* column.)

- **Make all document icons look like documents by including the folded-down corner.** Document icons on the Macintosh have been absolutely consistent in their use of the folded-down corner—except sometimes. The folded-down corner is an important term in the Macintosh vocabulary; people count on it and become confused when it is not there. (We have no interest in limiting developers' creativity, which is why we have reserved only this tiny symbol in the upper right corner, leaving the remaining acreage free.)

- **Make all file containers look like containers.** Currently we have several containers for files: folders, hard disks, soft disks, and trash cans among them. File folders should always have the file-folder look, physical-device icons should look as much as possible like the actual physical device, and real-world analogs should look like their real-world counterparts. In the future, I hope, we will have new and interesting file containers such as file cabinets, books, libraries, and vaults. When it will happen I don't know (probably not until more developers demand it), but when it does, these should also resemble their real-world counterparts.

- **Make all application icons look different from documents and file holders.** Given strict adherence to the standards for document icons and file containers, I am in favor of unleashing full creativity in the realm of application icons. As long as we adhere to the rules for documents and containers, the application rule for users is simple: If it doesn't look like either a document or a container, it's an application.

I disagree with you, Richard, in that I think it important to offer developers a little advertising space, in the form of their application icons. In fact, I look forward to the day when we will give developers some control over their window appearance too, by

Continued on page 25

By Bill Hollifield, Union Carbide

I Hate "Install"

I buy a lot of software. More and more, it comes with an Install program. Every time I press "Install", (even Apple's!) I get nervous. I'm nervous because I don't know what's going on and the software manuals and Readme files don't tell me. I'm no novice; I run a fair-sized network with some sophisticated capabilities. So far I've always seen "Installation Successful." But in my worst nightmares, I see another dialog box...



Here in Bound Brook Plant Engineering, Union Carbide, our Mac Network and software are top-of-the-line and our systems are pretty complex. We have big screens, accelerators, protocol converters, NetSerials, SUM, Mail, screen savers, Daynfiles, and so on. We use a number of inits. (While we aren't "init-crazy," several are necessary to our jobs). Because of this complexity, installing new software is a concern.

I'd Rather Deal with Inits. Basically, I don't like "Install" programs. I'd rather create the appropriate folders and drag the files to them. Installing a Mac program shouldn't be any more complicated than that! I understand the necessity for "Install" for System Software upgrades but it should be minimized everywhere else. I prefer Init/Cdevs to resources that have to be installed in the System file. I don't want any program that messes with my hard disk or System file as part of some copy-protection scheme. If you must impose an "install" program on me, I want to know exactly what it's doing and I want to know about possible conflicts. How do I know that your Install may not clobber some other resource that your "De-install" won't fix? Rather than dealing with Install, I'd prefer to handle init conflicts, or work out a way to live with the ones that can't be fixed with current software versions. Generally, with inits, I can quickly troubleshoot the various combinations—"Restart" is

much faster than an "Install/De-Install" cycle.

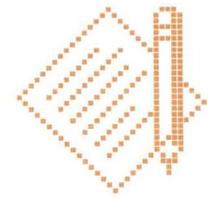
The trend, though, seems to be going the opposite way from what I would prefer. I just got a software update that requires "de-installing" the old version with the old version disk and then "installing" the new. Anxiety doubles! Sometimes, all an Install program does is combine pieces that won't fit on an 800K floppy. Why doesn't the documentation say that?

Developer, Save My Files. We upgrade our software all the time. The problem with updates is that the instructions generally assume that you don't have the prior version. I want to know what can be kept and what can be thrown away. For example, when a old user dictionary has a different icon than the new one, I assume that the Creator and Type codes may be different. Will the old files work? The documentation doesn't tell me and I'm tired of teaching word processors my name every time I upgrade. All upgrades should support the previous version's documents. We recently got a communications product upgrade that didn't, and had to rebuild some settings documents from scratch!

Here's another recent headache from a "Popular Cad" Readme file: "IMPORTANT NOTICE: To avoid the possibility of corrupting important System resources, please remove the Background Printing and Print Monitor files from your system folder before plotting with 'Popular Cad'." Are those named files the "important System resources?" Or will I corrupt resources (you

Continued on next page

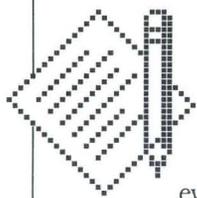
.....
 Bill Hollifield is the Plant Engineering Manager at Union Carbide's facility in Bound Brook, New Jersey, where there are 400 Macs. He's been a Mac enthusiast since 1985.



The Mac network and applications that I dream about will only happen if administering them remains an easy task.

Dear Developer

Continued from previous page



know—the things that ResEdit messes with) within my System file itself? And why the heck should plotting (in this example) mess up these other files?

And when exactly will it occur—when I configure the plotter or when I actually try to plot? And, can I ever put them back?

Still another example—I recently installed “Popular Page Layout” on a Mac IIci. First, I had to restart in Finder (argh!). Then, the Install program began asking for the other floppies *but a bug rendered the dialog boxes plain white!* Actually, I guessed that floppies were being asked for and that turned out to be correct. Imagine what this would have done to a less experienced user—they would have thought the machine had crashed (that’s what it looked like) and turned it off.

Before my group gets new software, I (or another knowledgeable department Mac user) read the installation instructions and Readme documents. I install the program on my machine and write out any special procedures (given our department configuration) and any special circumstances. The department secretary then does all of the other installations. I note any applicable problem “buried” in the documents, e.g. “change the init name on the machines with Radius screens when installing SAM.” Or “Be sure the machines getting Dos Mounter have Apple File Exchange 1.3.”

Minimize Complexity, Increase Sales. Our computing future is going to get more complicated. Our Carbide Internet is spanning many different locations. We anticipate more involvement with Ethernet, SQL, the Data Access Language, and other sophisticated enhancements. Frankly, the Mac networking that is going on in Carbide is far advanced over the PC networking because it has required minimum resources to build and maintain, but the complexity is getting a little out of hand. Developers can help make the building and enhancement of our future networks easier (and sell more of your products) if there is minimum anxiety required for installation. The same goes for the problems associated with hardware keys and serial number broadcasts. If my life is made easier administering your product, then I will buy more of it. And you can be sure that there are some products that I will never let in the door again.

I was happy to give up fooling with IBM’s method of installing code at start-up time and dealing with a highly cryptic and ever-growing CONFIG.SYS file that kept loading more and more files. It is now getting to where I have similar problems on the Mac. Sure, it’s still nowhere near what CONFIG.SYS for OS/2 is like, but we could (and have to) be better!

I realize that the developer’s task is made more difficult to the extent that the user’s is made easier. But the Mac network and applications that I dream about for the future will only happen if administering them remains an easy task. Let’s kill this budding DOS-type Install phenomenon while we still can! 🍏

It Shipped

The following is a list of companies that participated in the “It Shipped” Program between April 27 and May 31.

4-Sight—**QueFax**

Accolade—**Balance of the Planet**

Andromeda Computer Systems Ltd.—**Master Tuner**

BCI Boston Companies Inc.—**QuickShot Professional**

C-Cube Microsystems—**C-Cube Compression Workshop**

Cheshire Group—**PinPoint Error Reporter**

Computer Support Corporation—**On-Command**

Galactic Salvage, Inc.—**Legal Practice Made Perfect**

GDT Softworks Inc.—**JetLink Express**

Gizmo Technologies—**Send Express**

HyperGlot Software Co., Inc.—**Learn to Speak French**

Insignia Solutions, Inc.—**SoftPC- EGA/AT Option Module**

Koala Technologies—**MacVision**

Koyn Software—**IFS Explorer**

Lasec GmbH & Co.—**Archive One CD-Rom**

Leckie Associates, Inc.—**Cost/Schedule Workshop CSW**

Forecasted Estimate at Completion, Cost/Schedule

Workshop CSW Performance Measurement Data

Analysis, Cost/Schedule Workshop CSW

Responsibility Assignment Matrix

Micron Technology, Inc.—**XCEED IICI-128**

Microsoft Corporation—**Microsoft Excel DataAccess (CL/1 support)**

Neotech Ltd—**Neotech Image Grabber NUBus/SEBus Library, Neotech TimeFrame**

Northgate Computer Systems, Inc.—**OmniMac Ultra**

Novell—**TCPort Developers Kit**

Radius, Inc.—**The Pivot Display**

Ramsay Consulting—**telnet•PM**

Seven Hills Software Corp.—**Disk Access**

SNA, Inc.—**Stylist for Microsoft Word**

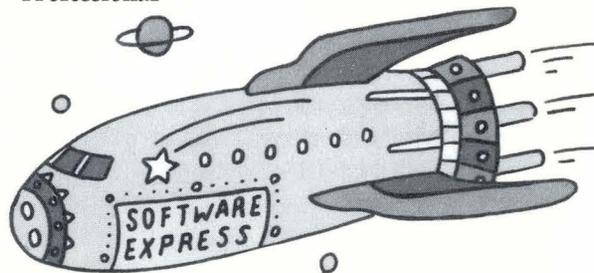
SUMBOLON CORPORATION—**Tiny Pim personal information manager**

Transitional Technology, Inc.—**Nightshift**

Tri-Data Systems, Inc.—**LanWay TR 16/4**

Vertical Solutions—**Fast Label 2.0, FastEnvelope**

Virginia Systems, Inc.—**Sonar Bookends, Sonar Professional**



By Jim Davis

Apple Sees Through Windows

Just in case you've been vacationing on a remote island for six months, read it here first: On May 22, Microsoft Corp. introduced Windows 3.0, a graphical user interface for MS-DOS computers. The announcement, of course, was watched by the entire computer industry and generated a great deal of press.

Many of the articles have positioned Windows against the Macintosh. They say that it's a GUI (graphical user interface) war that Microsoft will win because it is selling a "Mac-like" interface that can be used on any clone. They claim this erases the major differentiation between Macintosh computers and PCs. They further conclude that Apple will have a hard time selling Macintosh systems

But that's not the only story the press has discovered. There are also many articles that focus on the impact Windows 3.0 may have in the Intel-based world. If Windows 3.0 is successful, it will further delay adoption of OS/2 and Presentation Manager as the new industry standard. Finally, some articles and consultants suggest that both Windows and the Macintosh will increase in share over time, at the expense of plain DOS and OS/2.

This situation is reminiscent of last year at this time, when OS/2 and Presentation Manager were expected to take over the personal-computing world (just like NeXT before them). Given all the hubbub, how can you interpret these events for yourself (and your customers) as an Apple developer? Here are a few ideas to keep in mind:

Windows 3.0 further confuses the Intel-based world. Windows 3.0 is definitely an improvement over previous versions of Windows and will be valuable to many DOS users. It also affects the Intel-based world in several ways.

First, it will extend the life of DOS and therefore delay the adoption of OS/2 as the new Intel-based industry standard. Not only will Windows 3.0 increasingly cloud the issue of when users are supposed to move to OS/2, but it also adds yet another interface choice for DOS users. That's obviously a blow to IBM. DOS users already have many confusing choices to make, and Windows 3.0 adds another. It is one more DOS interface that customers will have to evaluate, in addition to several revisions of OS/2 with and without Presentation Manager, the two earlier versions of Windows, 50 or more DOS shells, and several versions of MS-DOS.

It also gives Intel-oriented developers one more major platform for which to develop, further stretching their development resources and slowing OS/2 application development as well.

Windows 3.0 should be viewed as a transitional product only—meaning that customer growth paths are uncertain. Will there be a Windows 4? Or will OS/2 come of age in the future, making current hardware and software obsolete?

Windows 3.0 moves the competitive debate to Apple's turf. Windows 3.0 is a milestone—it marks the passing of the command-line interface. Thus, Microsoft is forcing the market to compete on Apple's terms.

From Apple's perspective, we're pleased that the industry is paying so much attention to the graphical user interface. It's an endorsement of our leadership. When customers compare other graphics-based products with the Macintosh, Apple wins, because we have a six-year lead in graphics-based applications and technology refinement. When customers take a serious look at a graphics-environment computer, they find many advantages with the Macintosh (see below), which gives rise to the scenario that both Windows and Macintosh will gain market share, at the expense of plain DOS and planned OS/2 migration.

The Macintosh's differentiation goes far beyond its graphical user interface. There are many ways in which the Macintosh remains superior to Windows 3.0, but here are four of the major ones:

- *Only the Macintosh has a unified architecture.* Apple designs everything itself—hardware, system software, and built-in networking. In the PC world, the hardware and software standards are controlled in separate places—Microsoft makes the software, while the hardware "standard" is set by several hundred compa-



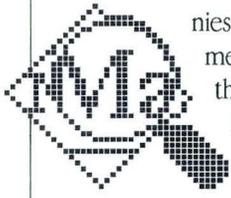
Genuine ease of use has to be designed into the computer from the chips up; you can't add it on.

Continued on next page

.....
Jim Davis is Apple's Director of Marketing, System Software.

Market Insight

Continued from previous page



nies that aren't able to agree on a single standard for memory expansion or the system bus, let alone anything more complicated. What this means to users is that the Macintosh is better integrated and that it more quickly incorporates innovations such as SCSI and built-in networking.

- *The Macintosh has a much smoother growth path.* All Macintosh computers—including the very first ones introduced in 1984—will be able to run the latest Macintosh system software, System 7.0, requiring only an increase in memory and storage. What this means to users is protection of their investments—in hardware, software, and training. In contrast, users with Intel-based machines face major hardware upgrades, which are both complicated and expensive, for Windows 3.0.

Most of the installed base of 20 million or so PCs use the 8088, 8086, or low-speed 80286 processors, which market analysts such as InfoCorp say won't be able to run Windows 3.0 at an acceptable speed. To move to Windows, those customers will face major hardware upgrades. And even then, they won't be through, because a couple of years down the road, it will be time to move to OS/2, and a lot more hardware will then be unusable.

One of the greatest misconceptions of the whole Windows 3.0 discussion is the notion that Windows will give Macintosh functionality to cheap clones. But a cheap clone cannot run Windows 3.0 without adding a lot of extras, so it's not cheap anymore. You have to add extras such as a high-resolution card, a graphics monitor, a hard-disk drive, a mouse, Windows-specific applications, and additional RAM.

- *The graphical user interface is only one of the reasons why the Macintosh is as easy to use as it is.* The Macintosh edge in ease of use won't change, because ease of use can't be added on as an afterthought. Apple builds it into everything it does—and that includes simple setup and configuration, easy integration of the Macintosh on networks with peripherals and CPUs from other vendors, and developer tools for creating consistent applications for users.

The main differentiator of the Macintosh is its superior functionality, and that has not been duplicated by the competition. If someone imitates a single feature such as the interface, it does not substantially reduce our overall lead in functionality. In other words, "It may look like a Mac, but it doesn't work like one." Genuine ease of use has to be designed into the computer from the chips up; you can't add it on later.

- *The Macintosh has a big lead in graphics-oriented applications.* Over the last six years, Macintosh has accumulated a library of thousands of applications—the largest library of graphically based applications in the world. The competition won't be able to match that quickly, if at all. What this means to users is the freedom of choice to pick the programs that work best for them.

When customers sit down and consider what they can actually do with the Macintosh, compared with what they can do with

MS-DOS and Windows, the advantages of the Macintosh are very compelling.

Look, for instance, at the recent user-satisfaction study by Diagnostic Research, Inc. Users rated the Macintosh higher than DOS and Windows in a long list of comparisons—overall satisfaction, versatility, ease of use, reliability, performance, purchase recommendation, and a lot more. The survey was conducted with Windows 2 users, but the fact is that underneath *any* version of Windows is MS-DOS, and that's the real issue. The main advantage of the Macintosh is its overall functionality.

Looking for Value. Take a look at what has happened in the auto market. Buyers are selecting the best value, not the cheapest. As a matter of fact, the best-sellers are well above the cheapest in price. If price were all that mattered in the Intel-based world, why would companies such as Compaq continue to grow while clones undercut their price significantly and run the same operating system and software?

We believe that as people begin to shop for value in computers, they will increasingly turn to Apple, because we provide value that others can't.

It's useful to remember that some new competitive product is predicted to crush Apple almost every year—OS/2, NeXT, SPARC-Station 1, GEM, Open Look, Commodore's Amiga...the list goes all the way back to the PCjr. The Macintosh has thrived, in spite of the competition, because it's an established computing standard and no single product can bump a standard aside. 🍏

Cover Story

Continued from page 14

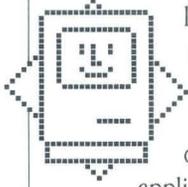
necessary to do hardware-based video compression and expansion (at least not for the next few years). By May of next year, Apple will deliver a software-only video-compression/expansion generalized solution for the Macintosh. This is yet another system-software-level technology that Apple will provide to make it easy for you to add another medium (video, in this case) to your programs.

"This Changes Everything." As Jonathan Seybold said at the beginning of this article, integrating new media—sound, voice, animation, and video—into our current conception of "program" and "document" *does*, in fact, change everything where personal computers are concerned. Two years from now, sound and voice will be as commonplace in programs and documents as multiple fonts are now. Animation and video will follow. And after that—who knows?

As a line grows to a square and then a cube, so does each new medium add a new dimension to the quality, expressiveness, and depth of the programs that use it. By putting the appropriate tools into system software, Apple will tremendously lower your "cost of entry"—in both time and money—into these new media (and prevent incompatibilities by providing a single hardware-independent model for using them). We hope that you will join us in the exciting future just ahead—but, for you as a developer, the future starts *now*. 🍏

The Human Interface

Continued from page 20



letting them design their own scroll-bar or title-bar appearance. (This will happen if and when enough developers demand it.)

Application icons must never use a folded-over corner, which would make users confuse the application with a document (nor should document icons incorporate the hand object). I find it a good practice to avoid an upright rectangular appearance as well, as it tends to trigger the "Gee, it must be a document" reflex.

- *Invent new objects, with new appearances, for new behaviors.* If you need to change grouping in a database, create a new machine designed to change the groupings in a database. Don't use the scroll bar! The scroll bar is not a Vegemetic designed to slice, dice, shred, and peel. It is a single-purpose tool, designed exclusively to make things scroll.

- *Devise menus that change to reflect the current mode of the system.* The most dramatic mode changes on the Macintosh occur when applications are switched. Few would argue that switching from WordPerfect to Excel should not cause the menu bar to change. If we just kept adding new menu sets, side by side, we would soon have the menu bar extending out 25 feet, requiring extremely wide, if not so high, monitors. But how about within applications? Is it OK to jump things around and shorten or lengthen menus willy-nilly? Of course not. But many applications have become too complex and too powerful to maintain a static menu structure.

So how do you change the menus without confusing your users? The strategy that works is to change the menu layout within an application when—and only when—users change their own mental mode. My favorite example of this occurs in SuperPaint, which maintains both a drawing and a painting layer. When a user has selected the Draw layer, the paint tools on the pallet are replaced by drawing tools and the Paint menu gives way to a Draw menu.

When SuperPaint had been out around a year, I began hearing complaints from human-interface types that SuperPaint was violating the guideline requiring that menus never change, a guideline I had always supported. Even though this menu switching had never bothered me personally, I decided to discover whether it was bothering any other SuperPaint users. I spent the better part of a day wandering around through Apple's nonengineering buildings doing a survey of experienced nontechnical SuperPaint users. The results were quite startling: More than 85 percent of these experienced users flatly denied that the menus changed at all. Most were quite insistent that I must be using a different version of the software, since on their machine, both the Paint and the Draw menus were always available.

What was happening here? Both menus were always available, in the sense that the users never went off in search of either one without finding it, since they had consciously changed their

own mode in the process of changing the application's mode. When the users decided to draw and let the system know, the universe filled with drawing things. When they decided to paint, everything to do with painting was instantly available. This menu switching worked, not only freeing up a precious column in the menu bar but also giving the users additional feedback about the mode by displaying the Draw or the Paint title in the menu bar.

The first goal of any application-design team should be to avoid menu switching. Only when that precious real estate has run out should such a scheme be considered, and then only when a strong user mode can be identified and subsequent testing shows either positive benefit or at least the absence of deleterious effects.

Finally, Richard, I disagree with your statement "In the end it's the users who say what interface gets accepted." Consumers tend to buy what their salesperson tells them to buy, and they too often are attracted to the most-difficult-to-use systems and applications in the erroneous belief that complexity means power and power implies machismo. It is up to all of us, Apple included, to stop telling people that the Macintosh and the applications that run on it "are so easy to use that even a total fool like you can do it!" They may have been easy to use in 1984, when our people were writing one-page memos, but now they're creating ten-volume books and one-hour videos. Instead, we should be telling them: "Our inherent ease of use has allowed us to build the most-powerful, most-productive applications on the market, applications that even a total fool like you can use!"

Or something like that.

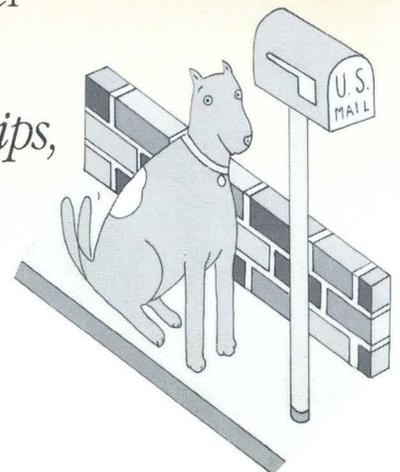
My Link address remains: TOG 🍏

Coming Soon

- MacApp Success Stories
- Network Application Tools
- Virtual User

PLUS...

Marketing tips,
Human
interface
issues,
and more.



By Jim Merritt

Parallel Development: Earn While You Learn

Today in the U.S., Apple II computers dominate K-12 education: Better than three out of every five computers in schools are Apple II or IIGs computers. We believe that the Apple II product line will remain a major platform for educational computing throughout the 1990s. Yet the graphic, desktop-style interface, first popularized by the Macintosh, is clearly the future of personal computing. You need only browse through recent industry magazines or scan the shelves at any software outlet to see evidence that the desktop interface idea has proliferated within and outside of Apple's sphere of influence. Although the K-12 installed base of the Macintosh is growing, the Apple II installed base continues to grow as well. This situation may pose a quandary for educational software developers.

On the one hand, the writing is on the blackboard: To win in education in the '90s, you must offer curricular software with a desktop interface. On the other hand, how can you move in that direction and stay afloat until sufficient numbers of systems in schools support the desktop interface? Most certainly, learning to program in a desktop interface environment isn't something you can do overnight, and time spent developing applications for other interfaces is time lost in mastering the Apple desktop metaphor. Yet for developers of educational software, writing nondesktop applications for Apple II series computers is one of the surest ways of putting bread on the table.

When asked how to reconcile the need to grow with the need to survive, I answer that you don't have to give up steady income potential to adopt the graphic desktop interface. Most schools have already invested in a computer system that fully supports it: the Apple IIGs. Whether you are an Apple II developer who wishes to learn desktop programming or an experienced Macintosh developer wanting to crack the Apple II-dominated educational marketplace with an application that features the desktop interface, you can succeed today with the Apple IIGs.

The Apple IIGs: Plentiful, and Plenty Capable. In less than four years, the Apple IIGs has made quite a mark on the computer industry. Better than one in seven Apple II computers *ever sold* sports an Apple IIGs label. The IIGs provides a rich toolbox that helps you endow your software with a colorful, responsive desktop interface

that follows Apple's human interface guidelines. The IIGs toolbox also helps you produce breathtaking sound, speech, and music through the built-in Ensoniq synthesizer chip as well as through MIDI, the Musical Instrument Digital Interface. Finally, the IIGs offers full toolbox support for the Apple II Video Overlay Card. With our tools, you can bring the increasingly important dimensions of video, sound, speech, and multivoice, multichannel music to your classroom products.

The IIGs also offers GS/OS, a powerful, efficient operating system that provides complete, transparent access to the AppleTalk[®] network and AppleShare[®] file servers—without requiring additional hardware. This means that well-behaved desktop interface applications, even those written for the Apple IIGs *prior* to AppleShare support, can now access network volumes as easily as disk volumes. The same operating system calls that apply to disks and disk files apply equally well to network volumes and files. As more schools connect their classrooms with networks such as AppleTalk, it will become essential for your products to work smoothly with disks as well as network volumes. GS/OS helps you satisfy that demanding requirement easily.

You can create Apple IIGs desktop interface applications on the IIGs itself, using the Apple IIGs Programmer's Workshop (APW). Those who prefer the Macintosh Programmer's Workshop (MPW) can use its MPW IIGs Cross-Development Suite. Both development environments are powerful and mature, providing many tools and features to expedite the production and maintenance of your software products. In addition, MPW enables you to use one machine to develop for the huge and still-growing installed base of Apple IIGs systems as well as the increasing number of Macintosh systems in K-12.

Jim Merritt is the Manager of the Apple II Applications, Utilities, and Development Environments group in Software Engineering. His business card identifies him as "Presbyte," a title that comes from a root meaning both "elder" (he started with Apple in 1980) and "someone who takes a longer view" (which he tries to do whenever possible). Before accepting his current assignment in 1988, Jim managed the Apple II Developer Technical Support group, but he has also earned a living as computer programmer, radio DJ, technical writer, magazine columnist, and ranch hand.

Parallel Development: Methods and Tools. Because the Apple IIgs and Macintosh share the desktop metaphor, toolbox architecture, and AppleTalk network, and because you can develop programs for both computers under MPW, it makes good sense today to undertake parallel development of essentially identical desktop interface programs for the Apple IIgs and Macintosh. Through *parallel development*, your intellectual investment in mastering the popular desktop interface can be spread across our two most important educational platforms. More importantly, your ability to field corresponding versions of your products for the Apple IIgs and the Macintosh may be the key to your long-term success in the education market.

Your intellectual investment in mastering the popular desktop interface can be spread across two important platforms.

In parallel development, you write the application-specific parts of your program in a high-level language, typically C or Pascal. To complete a version for any particular target machine, you must then write several system-dependent modules and re-define several data structures especially for the target. If the target systems lack built-in toolboxes, writing the system-dependent modules and data structures can actually turn out to be the bulk of your development effort. But when developing in parallel for the IIgs and Mac, you can exploit the similarities of the two systems and their toolboxes to reduce the amount of system-specific code and data you must produce. Today you can also use "power tools" to expedite creation of the system-dependent pieces in a minimum amount of time.

Programs for the Apple IIgs, such as Design Master from the Byte Works and Genesys from SSSI, Inc., permit you to prototype and experiment with your interface on the IIgs itself. When you're satisfied with your design, these programs produce a combination of resources and source code in any one of several languages (including assembler, C, and Pascal) that you can then integrate with the generic portions of your program to produce a completely functional desktop application.

For those who want to use the Macintosh as their development system, AppMaker and AppMaker/GS, from Bowers Development Corporation, allow you to prototype your basic interface design on the Mac and then generate shell programs to realize your design for either the Macintosh or the Apple IIgs. Finally, to get model source code that demonstrates how to write both the generic and the system-specific portions of your programs, contact Apple's Developer Technical Support group (AppleLink or MCI Mail: AIIDTS).

Two Products, but Not Twice the Effort. The author of Design Master, C.K. Haun, estimates that experienced desktop interface programmers will need only about 15 percent more time and effort to produce the Mac and GS versions of an application together than to write either version alone. This assumes that both packages are written from scratch and that parallel development is pursued from the outset. Much of the additional effort will go into version-specific design and coding to emphasize the strengths and mitigate the weaknesses of each platform. For instance, the highly detailed graphics and icons that are suitable for the Macintosh version may need to be reworked for the IIgs to allow for the latter's different video resolution. On the other hand, multivoice, stereo musical arrangements that are ideal for the IIgs may need reworking for the Macintosh to provide for the Mac's different sonic capabilities.

Today's parallel-development tools can also eliminate much work for Macintosh developers who want to convert a Macintosh application to the IIgs. In particular, AppMaker/GS can extract the resource definitions of such things as windows, menus, and dialog boxes from your existing Macintosh application and then translate them into resources for the Apple IIgs.

Those who are not familiar with the desktop interface or toolbox programming may find that developing an Apple IIgs product provides for a gentler introduction to the methodology, since the IIgs toolbox automates more of the mundane chores associated with the desktop interface than does the Macintosh toolbox. But doing either the IIgs or the Mac versions alone practically invites you to write code that will prove incompatible with the other system. Doing both versions together will help you keep them in sync.

Apple IIgs + Macintosh = Sustained, Long-Term Income. We will continue building, selling, and improving the Apple IIgs in the foreseeable future, so the already large Apple IIgs market will continue to grow. And as more Macintosh computers appear in K-12 classrooms, you'll be able to sell more copies of your Macintosh versions, especially to those customers who are happy with your IIgs editions (or who know someone who is).

For Macintosh programming veterans, parallel development of Apple IIgs and Macintosh applications gives you the opportunity to use your desktop interface knowledge to open new markets and generate new revenue without much additional engineering effort. For newcomers to desktop interface programming, parallel development gives you an unparalleled opportunity to earn while you learn. 🍏



GetNextEvent

The "🍏" indicates the trade shows/events at which Apple Computer, Inc. will exhibit. This list may be incomplete. If you have information about a show that you want listed here, contact Diane Wilcox, 20525 Mariani Avenue, Mail Stop 75-3B, Cupertino, CA 95014, Link: WILCOX.DM. For further information check the Events folder on AppleLink (path: Developer Services:Events).

Date	Show	Location	Contact	Phone/Link
7/17-7/19	Macintosh/New York	New York, NY	Cambridge Marketing	617/290-0400
7/20-7/21	A-2 Central	Kansas City, MO	Sally Dwyer	913/469-6502
7/29-8/1	🍏NUL—Nat'l Urban League	New York, NY	Miriam Morales	212/310-9037
7/31-8/2	🍏AAAI—Amer. Assn. for Artif. Intelligence	Boston, MA	Steve Taglio	415/328-3123
8/7-8/9	🍏SIGGRAPH	Dallas, TX	Barbara Voss	212/752-0911
8/8-8/11	🍏Macworld Expo	Boston, MA	Mitch Hall & Assoc.	617/361-8000
8/12-8/15	UPCAEDM—Univ. Programs in Computer Aided Engineering, Design, Manufacturing	Ann Arbor, MI	Joe Easley	313/764-3334
9/4-9/8	Exhibition of Business Machines & Equip.	Helsinki, Finland	Juha Usva	Link: USVA1
9/4-9/8	🍏SwissData	Baal, Switzerland	Corinne Suter	Link: IT0006
9/6-9/9	🍏CyberArts International	Los Angeles, CA	Miller Freeman Expositions	415/995-2471
9/14-9/16	AppleFest '90	San Francisco, CA	Cambridge Marketing	617/290-0400
9/17-9/19	🍏FCC—Federal Computer Conf.	Washington, DC	Information Development Corp.	301/961-8990
9/17-9/19	🍏Presentations '90—Graphics & Multimedia	Los Angeles, CA	Cambridge Marketing	617/290-0400
9/19-9/22	🍏Apple Expo	Paris, France	Fabienne Roch	Link: FRA.PROMO
9/19-9/26	Bureau '90	Brussels, Belgium	Veerle van der Eecken	Link: VANDERECKEN
10/3-10/5	🍏Seybold—Electronic Publishing	San Jose, CA	Kathleen Kaiser	213/457-5850
10/4-10/6	🍏Macworld Expo	Sydney, Australia	Macworld Expo	Link: AUST0261
10/4-10/8	SMAU—Info. Telematics & Office Prods.	Milan, Italy	SMAU	39-2- 78 17 09
10/14-10/17	Educom '90	Atlanta, GA	Lynn Miranda	609/520-3350
10/18-10/20	🍏Closing the Gap '90	Minneapolis, MN	Mary Ann Harty	612/248-3294
10/22-10/26	🍏Systec—Computer-Integrated Manuf.	Munich, Germany	MMG	089-51070
10/25-10/30	🍏Orgatec Koln Int'l Office Trade Fair	Cologne, Germany	Koln Messe GmbH	0221-8210
11/4-11/7	🍏SCAMC—Symposium on Computer Applications in Medical Care	Washington, D.C.	Greg Thomas	202/994-4285
11/12-11/16	🍏Comdex	Las Vegas, NV	Interface Group	617/449-6600
11/13-11/15	🍏Autofact	Detroit, MI	SME	313/271-1500
11/27-11/30	CAUSE '90—College & Univ. Systems Exch.	Miami, FL	Debbie Smith	303/449-4430
1/10-1/13	🍏Macworld Expo	San Francisco, CA	Mitch Hall & Assoc.	617/361-8000
1/22-1/24	🍏Uniform	Dallas, TX	PEMCO	800/323-5155