

## Inside this issue

### TECHNOLOGY & INDUSTRY

DEVELOPERS SHOWED more than 100 QuickTime-compatible applications at the Macworld Expo earlier this month, including mainstream applications that have added QuickTime support as well as new classes of products optimized for QuickTime. Apple is educating its field sales force about QuickTime applications via an electronic version of Apple Intro News.

See the News Folder

APPLE HAS JOINED the Business Software Alliance, an organization set up to protect U.S. software copyrights throughout the world. Other BSA members include Aldus, Ashton-Tate, Autodesk, Digital Research, Lotus, Microsoft, Novell, and WordPerfect.

See the News Folder

THIS MONTH'S CD features several intriguing new bits of sample code as well as the new Network Software Installer 1.1, and new testing and debugging tools. You'll also see that the Developer Info Assistant (DIA) and Search Knowledge Base have been merged, with the DIA now serving as the main search facility for the CD.

See the News Folder

THE QUICKTIME Developer's Kit is now available through APDA, as are new versions of the Macintosh Communications Toolbox, the Macintosh Programming Fundamentals self-paced course, and MIDI Management Tools.

See the News Folder

FIRST THE COLUMN, now the book. Look for Bruce Tognazzini's new book from Addison-Wesley—Tog on Interface—in a bookstore near you. The basis for Tog's book are his Apple Direct columns, as well as your inquiries to him on interface issues.

See the Tog Folder

IF YOU'RE OUTSIDE the U.S. and want to buy APDA products, you can choose to work with local APDA programs. This month, we provide the most current list of international APDA contacts.

See the News Folder

GET SOME HELP from the range of developer resources available to you, including developer associations, the Macintosh Services Directory, local DU training, and more.

See the News Folder

VIRTUAL USER V. 1.1 and the Apple Publications Style Guide were recently released by APDA.

See the News Folder

## **BUSINESS & MARKETING**

**ON TECHNOLOGY** has had plenty of experience creating site license and volume-purchase agreements. John Shagoury, On's vice president of sales and marketing, shares his advice on how to create a win-win site license.

See the Business and Marketing Folder

**THE 10 MOST COMMON** product launch mistakes have been observed again and again by Leigh Marriner, a consultant who specializes in competitive strategies for marketing companies.

See the Business and Marketing Folder

**APPLE'S UNIT SALES** have been doing quite well, thank you very much—a fact that's been confirmed by Audits & Surveys in a recent dealer channel study. Apple's unit sales were up 61 percent from year to year, while the industry overall was up only 11 percent.

See the Business and Marketing Folder

**MARKETING TIPS** from dozens of marketing executives are captured in the new Do's and Don'ts of Software Marketing, from the Software Publishers Association.

See the Business and Marketing Folder

## **Tools Advisor from DU**

Wondering which development tools would be best for you? The Macintosh Development Tools Advisor can help you find the answers. The Tools Advisor offers comprehensive technical data on more than 80 programming tools: compilers and languages, debuggers and prototypers, CASE tools and multimedia packages, all in a rich hypertext system.

It also includes several essays on critical topics such as object-oriented programming, Apple events, and System 7. The Tools Advisor is a HyperCard stack developed by Apple's Developer University and is scheduled for release this month.

In preparing the Tools Advisor, Developer University wanted to go beyond a list-oriented catalog of tools and give you information about how the tools get used in actual projects. The result was to include a collection of stories by experienced programmers. These developers describe their experiences and give advice and cautionary tips on the use of the tools. To help you find stories most appropriate to you, the Tools Advisor allows you to define a loose profile of your needs and then searches for stories from developers with similar backgrounds and tasks.

The Tools Advisor also includes a glossary of technical and trade terms. Glossary entries and cross-references give additional support while you navigate through the intricate terrain of technical information.

Two versions of the Tools Advisor are available. The disk-based edition includes screen shots and text descriptions. The CD-ROM edition of the Tools Advisor adds demonstrations of dozens of tools. You can, for instance, take a multimedia tool for a test drive as you learn about animations and other developers' experiences with that particular product.

The Tools Advisor is available through APDA. The price is set at \$10 for the disk edition (R0124LL/A) and \$15 for the CD-ROM edition (R0171LL/A). (See "Now Available from Apple" on page 2 for ordering information.)

If you receive the monthly developer mailing, you will receive a copy of the disk edition on the Developer CD. To use it, you need a Macintosh with System 6.0.5 or later, HyperCard 2.0 or later, and a hard drive. A CD-ROM drive is necessary to use the Tools Advisor CD.

### **AppleLink 9600-Baud Access**

Last month Apple released AppleLink 9600-baud service in the United States in 78 cities. AppleLink 9600-baud service is a premium service that provides greater speed, increased productivity, and substantially reduced costs for downloading files.

Users who want the maximum speed for all activities will use 9600 baud.

However, if you want the lowest cost for services that are not data-intensive, such as messaging and browsing, you should continue to use 2400 baud. AppleLink 2400-baud service is currently priced in an hour and kilocharacter charge combination, which usually results in lower costs for non-data-intensive activity.

The AppleLink 9600-baud service is priced at a flat hourly rate of \$37 per hour, effective December 1, 1991. Charges will first appear on the January 1, 1992, AppleLink usage bills.

A 500K file downloaded from AppleLink at 2,400-baud currently costs a user about \$35. With 9600 baud, the cost will be about \$8, a 77-percent savings!

Access nodes for 9600-baud service are not yet readily available throughout Europe and the Pacific region. Apple is currently beta-testing 9600 baud in several countries and expects the service to be available internationally later in 1992. Watch Apple Direct for information about availability, cost, and pricing structures outside the U.S.

For more information on using 9600-baud service, including a list of access phone numbers and compatible modems, look on AppleLink in the AppleLink Guide bulletin board (path—AppleLink Information:AppleLink Guide:Premium AppleLink services:9600 bps access to AppleLink).

## **Opening the Net**

### **Making your programs network-savvy**

**by Gregg Williams, Apple Direct staff**

The number of Macintosh networks continues to grow, and with it grows the potential for new network-based products that make users more productive. Of the many products introduced last October (see "The October Revolution," in the October 1991 issue of Apple Direct), two of them present new possibilities in the fields of networking and remote communications.

Apple is making the architectures of both AppleTalk Remote Access and AppleShare Server 3.0 available to interested developers. In this article, we'll take a technical look at both products and examine the new tools they put at your disposal. They may very well suggest some exciting new development possibilities.

#### **AppleTalk Remote Access**

Stated simply, AppleTalk Remote Access lets a Macintosh connect to a remote Macintosh or AppleTalk network over standard phone lines. (This fact is of particular interest to Macintosh PowerBook users, who get AppleTalk Remote Access free with each PowerBook.) Since it is built on top of System 7 File Sharing, it requires that both Macintosh computers be running System 7.0 or later. The AppleTalk network on the remote end can be equipped with Apple's LocalTalk cabling, Ethernet, or a token-ring network.

Although AppleTalk Remote Access was designed to be an easy-to-use solution for an individual Macintosh user, Apple engineers designed the underlying technology to be modular and extensible. (The current AppleTalk Remote Access product works only with modems and a telephone connection, but this technology can work with other kinds of remote connections too.) The rest of this section will give you an overview of this technology and alert you to several items you may need to know about.

This is ARAP. The technology mentioned above is named, not surprisingly, the AppleTalk Remote Access Protocol, or ARAP. It is a communications protocol that allows a Macintosh to connect to a remote AppleTalk network.

(Strictly speaking, the AppleTalk Remote Access Protocol provides access to a remote network; the services on the network determine how users interact with it. The Finder, for example, displays the remote network's volumes and allows users to manipulate them as icons, folders, and windows.)

A new manager, the Remote Access Manager, handles the details of interacting with remote networks.

The AppleTalk Remote Access Protocol allows a Macintosh to connect to a remote network, using an arbitrary point-to-point link (PPL) that has the following characteristics:

- The link must send and receive data in packets of at least 604 bytes.
- The link is responsible for detecting and discarding packets with errors (that is, it may deliver only valid packets to the AppleTalk Remote Access Protocol).

- It may be desirable for the link to deliver packets in the same order in which they were sent, guarantee that the packets were received, and never send the same packet more than once (this is called a reliable link).

In such a case, the AppleTalk Remote Access Protocol can increase the effective speed of data transfer by doing "smart buffering," an algorithm being patented by Apple that speeds the transmission of duplicated data by replacing repetitive headers and duplicate packets with (shorter) tokens.

The AppleTalk Remote Access Protocol works with two kinds of packets: internal messages, which do various "housekeeping" functions related to the data transfer, and AppleTalk packets, which contain the actual data that is sent over the network.

The first (and current) implementation of the AppleTalk Remote Access Protocol assumes the use of a normal telephone connection and 2,400-baud (or faster) modems on both ends. It implements a reliable link with a software module called the Modem Link Tool.

This tool uses the Microcom Networking Protocol (MNP) to implement a reliable point-to-point link over an asynchronous serial connection—in this case, a Macintosh serial port.

To ensure that all remote-access programs work together, Apple is proposing that third-party developers use the AppleTalk Remote Access Protocol when creating new products that need remote AppleTalk connectivity.

To make the AppleTalk Remote Access Protocol more compliant with industry standards, Apple has been working with developers to enhance the protocol to work over PPP (Point-to-Point Protocol). Doing so will open the doors for developers who create products for multiprotocol environments.

Currently, the Apple/IP Working Group, part of the Internet Engineering Task Force (IETF), has a tentative standard (called AppleTalk Remote Access/Point-to-Point Protocol) out for comment. Apple and several major vendors have pledged to support the final standard.

Interacting with ARAP. Apple is making the AppleTalk Remote Access API (Application Programming Interface) available through APDA; see the "Products from APDA" sidebar on page 6 for details. You can use the AppleTalk Remote Access calls to:

- Create and terminate remote connections, either directly or through something called a connection document.
- Display a standard user interface for the creation and termination of remote connections.
- Mediate access to the serial port when AppleTalk Remote Access is "listening" for an incoming call but other programs need the same port.

Table 1 lists the main commands of the AppleTalk Remote Access API. Most of them are accessed by the same call, PBRemote Access, and the appropriate parameter block.

When using the Connect command to establish a remote connection, your program can build the parameter block within its code or it can include an FSSpec (file-system specification record) that points to a connection document. (A connection document contains all the data the Connect command needs in order to make the connection.)

Users can create connection documents with the Remote Access application that comes with the AppleTalk Remote Access product.

Connection documents offer several benefits to you and your program's users. First, they make it easy for the user to make the desired connection. Second, they make your code simpler than it would have been if you had specified the connection by building a parameter block.

Third and most importantly, connection documents make your program, today, compatible with future types of remote connectivity. Connection documents essentially isolate the details of the desired connection from the rest of your code. This means that if someone implements a new kind of remote connectivity (and the associated connection document), your program can use that document to make a remote connection without knowing—or caring about—how the connection is made.

**Working with Network Transition Events.** With AppleTalk, when the configuration of the network changes, some programs running on the Macintosh want to be notified. If they aren't and they try to access a service that is no longer available, the Macintosh can hang for several minutes until AppleTalk “times out” and decides that the service is no longer available—not a good example of human-interface behavior.

To implement the removal of services more gracefully, the AppleTalk Manager creates and maintains the AppleTalk Transition Queue, a queue of programs that want to be notified whenever an interesting network event occurs. Apple engineers realized that AppleTalk Remote Access should also let these programs know when it cuts a connection to a network.

AppleTalk Remote Access makes sure that this happens by sending a network transition event to the AppleTalk Transition Queue whenever it connects to or disconnects from a remote network.

**Serial-Port Arbitration.** If this implementation of AppleTalk Remote Access is set to monitor the serial port for incoming telephone calls, how are other programs on the same Macintosh going to use the serial port? Fortunately, a part of the AppleTalk Remote Access software called the Serial Port Arbitrator handles this task for all serial drivers that are registered with the Communications Resource Manager.

If serial-port arbitration is available and no other program on the same Macintosh is using the serial driver, Remote Access uses the serial port to passively “listen” for an incoming telephone call. When another program asks for a port by using the OpenPort call, the Serial Port Arbitrator allows that program to use the serial port. When the program calls ClosePort, the Serial Port Arbitrator lets Remote Access passively use the serial port again.

This method of serial-port arbitration may lead to difficulties for a very small number of Macintosh programs, namely desk accessories that open and close the serial port and other programs that do so repeatedly.

In the case of desk accessories, the incompatibility arises because they use a process ID to close the serial port that's different from the one they use to open it. (AppleTalk Remote Access depends on seeing the same process ID for both the opening and the closing operations.) The solution is for the programmer to create a faceless background task (a program that has no human interface) and have the desk accessory ask it to open and close the serial port.

Programs that open and close the serial port more than once don't work correctly because AppleTalk Remote Access believes that the serial port is free once it sees the first close operation. There's no way around this—such programs must be redesigned. Fortunately, very few programs have this problem.

**My Number or Yours?** One aspect of AppleTalk's ease of use is its plug-and-play nature. You don't have to be a rocket scientist (or, in real life, a network administrator) to plug everything in and have the whole network available immediately.

Nodes, for example, are assigned numbers "on the fly" as they connect to the network, so there's never any chance of conflict. (Assigning such numbers is a classic example of a network-administration task that humans really shouldn't have to do.)

Unfortunately, when you contemplate connecting your network to, potentially, any other AppleTalk network on the planet, there's a problem. Occasionally, the remote network you want to connect to has the same network number as yours. (Network numbers are assigned arbitrarily—often by an administrator who configures a router connecting two networks to each other. Once assigned, a network's number does not change.)

Fortunately, Remote Access has a way around this, one that guarantees that you can always see everything on the remote network and that it can always see you. (The latter consideration is trivial, since to the remote network, your Macintosh "looks like" another network node—so you're assigned a unique node number.)

When AppleTalk Remote Access connects to a remote network that happens to have the same network number as yours, it maps the remote network to a reserved value that is guaranteed to be different from that of any existing network.

This way, the two networks will be able to "see" everything on the other side.

(AppleTalk Remote Access automatically takes care of the mapping for you.)

However, if your program is doing sophisticated network management and is storing network addresses as data, you may need to use a remapper procedure. You can get a pointer to this procedure by calling the `GetCodeHooks` command within Remote Access; see table 1.)

**But Wait, There's More!** Even if you have no conflict with the remote network, you may still have a conflict with the network number of another zone on the same remote network.

(Strictly speaking, each zone is itself a network, and each is connected to other zones by router hardware. The entire collection of all connected zones is called an internet.)

Each zone has its own collection of network numbers, and a network number on the local network may conflict with a network number on the remote network.

When there is a network number conflict, there is no way that AppleTalk can access the network services in both networks. AppleTalk Remote Access always defaults to letting you see your own network, even though that means that items in the remote network with the conflicting network number become invisible to you.

Because an implementation that occasionally will not show you all of the remote network is not a good example of network design, Apple engineers devised a way to get around the problem. When you use the `Connect` command in the AppleTalk Remote Access API, you can set a `guaranteedAccess` flag that essentially cuts you off



from your network and shows you the entire remote network. (Actually, you can still see network items that are directly connected to your Macintosh—that is, items that do not connect to you through a router.)

Using this command, your program can get access to any file on the remote network to which you are connected.

On the user's side, checking the Remote Only box in the Network control panel is the user's way of guaranteeing access to all remote locations. ("Remote Only" might more correctly be named "Remote Network Only.")

This mode keeps the user's Macintosh from seeing anything on the local network—hence the name. However, the user's Macintosh can see non-networked printers that connect directly to the serial port, like the StyleWriter and the ImageWriter.

**Developer Alert.** Before you decide to create an AppleTalk Remote Access-compliant product, you should know a bit more about what's ahead for the technology.

The current implementation of AppleTalk Remote Access (see figure 1a) is just the beginning. It uses the Microcom Networking Protocol to make the point-to-point protocol reliable. The AppleTalk Remote Access Protocol sits on top of that.

If you intend to support the Macintosh community only, the current implementation is for you. You can start by ordering the AppleTalk Remote Access Developer's Toolkit; see the "Products from APDA" sidebar on page 6. (Remember that every PowerBook Apple sells comes with AppleTalk Remote Access, so there's quite a market there and it's going to keep growing.)

If, however, you want to develop products that support multiple protocols—AppleTalk, TCP/IP, and DECnet, for example—then you may want to support the version of AppleTalk Remote Access currently being worked on by the Internet Engineering Task Force, ARAP/IPP (figure 1b). With it, you can support a variety of networking protocols over PPP, including AppleTalk via the AppleTalk Remote Access Protocol.

This is not to say that you must choose between the current and next implementations of the AppleTalk Remote Access Protocol. You can write code for the current AppleTalk Remote Access and be able to use a sizable portion of that code to implement ARAP/PPP.

### **AppleShare Server 3.0**

For users, AppleShare Server 3.0 means more power and flexibility—a network of as many as 120 simultaneous users, spooling of as many as 5 AppleTalk printers, better control of file and folder sharing, and more. But AppleShare Server 3.0 means even more to you as a Macintosh developer. In essence, it gives you the chance to create and sell innovative AppleTalk-network-based products. Network users have long been clamoring for more network services than Apple could ever provide, so AppleShare Server 3.0 gives you the "hooks" you need to satisfy that market.

Of AppleShare Server 3.0's many features (also covered in last October's Apple Direct), two are of particular interest to developers such as you:

- Translation of the File and Print Servers to being "well behaved" System 7 programs that can share the Macintosh with other programs.

- New functions that allow the program to monitor and change network events (including more-sophisticated security services).

AppleShare Server 2.0 pretty much dominated the Macintosh on which it ran, making it unsuitable for most other uses (services). In contrast, AppleShare Server 3.0 opens many new possibilities, because of its ability to coexist with other programs on the same Macintosh. Apple expects that users will still dedicate a single Macintosh to serve a network, but now they can run other network-related programs, including the ones you can now make.

The rest of this section covers the new AppleShare Server 3.0 calls and how you might use them.

**Now Serving....** One AppleShare service that many developers such as you have requested is the ability to take some action when your program detects a specified AppleShare event. AppleShare Server 3.0 provides this in a mechanism called server event handling.

With server event handling, your program can notify AppleShare Server 3.0 of your desire to have certain code executed when a given event (or type of event) occurs. The routine of yours that executes when the given criteria are met is called the server event handler.

For various reasons of timing and resources, the server event handler should be very brief and just dump a buffer of data. Later, in the main event loop, your program should check the buffer at nullEvent time and do detailed processing when it sees a new buffer entry. See figure 2 for details.

(Because the server event handler may execute at interrupt time, you should spend as little time as possible in it before returning and you should not make any memory-manager or file-system calls.)

**Say Something!** As any shy person can tell you, you can do only so much by listening. In the case of AppleShare Server 3.0, the next step up from listening to AppleShare events is being able to “say” something and make AppleShare do something for you.

AppleShare Server 3.0 allows you to do this through its server control calls, which are summarized in table 2. These calls enable your program to monitor and control the major functions of AppleShare Server 3.0 file serving.

(Note: AppleShare File Server 3.0 is a superset of System 7 File Sharing, as are the associated control calls for each. If a Macintosh has System 7 File Sharing enabled and someone installs AppleShare Server 3.0, the latter takes precedence over the former and converts certain data files to AppleShare 3.0 format; the change is permanent.)

**Users and Groups Calls.** As we’ve seen above, AppleShare Server 3.0 lets you listen to server events and issue commands to the server. The only pieces needed to give you meaningful control of AppleTalk networking are access to and control over the user and group entities that are common to both AppleShare Server 3.0 file serving and System 7 File Sharing.

The relevant information is held in one file, the Users & Groups Data File, and AppleShare Server 3.0's Users and Groups calls allow you to access and change their data. Table 3 summarizes the Users and Groups calls.

### All Together Now

With AppleTalk Remote Access, you can help your customers decrease the "pain" of being away from their primary Macintosh by allowing them to connect to remote volumes and services. The enhancements you might make can be anything from the simple (letting your program print on a AppleTalk printer on the remote network) to the complex (having your program do different work based on what it finds on the remote network).

With AppleShare 3.0's server event mechanism, server control calls, and Users and Groups calls, you have the tools you need to create virtually any AppleTalk-related product, software or hardware, that you can imagine. Potential products include those that do the following:

- Extensive auditing and tracking of AppleTalk networks (for example, keeping track of what files are getting downloaded);
- Better automatic backup of servers;
- Remote administration of servers from anywhere on the network
- Sophisticated network administration and control without human intervention, such as automatically logging off users who have had no activity in, say, 30 minutes).

As always, we expect developers such as you to invent powerful new products that will surprise and delight us as much as they will your customers. More and more people are using their Macintosh computers to interact with both human and computer resources beyond the walls of their own Macintosh. If your products help them do that, you have a better chance of succeeding in the competitive years ahead.

**Table 1:  
Commands available from AppleTalk Remote Access calls**

<b>Command</b>	<b>Function</b>
Load	Loads the Remote Access Manager into memory (if it is not already present)
Unload	Unloads the Remote Access Manager, frees it memory for reuse
Connect	Initiates an outgoing connection
Disconnect	Ends an existing connection or one in the process of being created
IsRemote	Determines whether a network address is local or remote
Status	Obtains information about remote access (for example, how long a connection has been active, how much time remains)
MungePW	Encrypts a password that is about to be stored in a document
GetCodeHooks	Returns a pointer to the remapper procedure, which remaps network-number/node addresses in programs that pass network addresses as data (most programs don't need this)

(as of December 1991)

**Table 2: AppleShare Server 3.0 server control calls**

<b>Call</b>	<b>Function</b>
SCStart Server	Starts up the Server
SCAlertEntity (internal use only)	Sends a high-level event to the specified entity
SCShutDown	Shuts the server down
SCCancelShutDown	Cancel a shutdown or disconnect command
SCDisconnect	disconnects specified users from the server
SCPollServer	Returns information on what the server is doing
SCGetExpFldr	Gets information about a certain shared volume, including its name, location, and the number of users who have mounted the specified folder
SCGetSetupInfo	Get the server setup data
SCSetSetupInfo	Set the server setup data to specified values
SCSendMessage	Send a server message to the specified users
SCServerStatus	Get selected server status data
SCInstallServerEvent Proc	Installs a server event procedure
SCRemoveServerEventProc	Removes a server event procedure
SCGetServerEventProc	Gets the head of the server event queue
SCServerVersion	Gets server version information
SCSyncWithInit	Called by the server to “wake” an INIT to allow it to inspect some change that has occurred (internal use only)
SCSetCopyProtect	Sets the copy protect status of a file
SCClrCopyProtect	Clears the copy protect status of a file
SCDisconnectVolUsers	Disconnects all users that have mounted a specified volume
SCGetUserMountInfo	Returns data on user’s usage of a specified volume
SCWakeServer	Wakes up the server
SCSleepServer	Used to shut the server down temporarily

**Table 3: AppleShare Server 3.0 Users and Groups calls**

**Calls relating to the Users & Groups Data File**

<b>Call</b>	<b>Function</b>
UGOpenFile	Opens the Users & Groups Data File at the specified location on the network
UGCloseFile	Closes an open Users & Groups Data File
UGCreateFile	Creates an empty file named “Users & Groups Data File”

**Calls relating to users**

<b>Call</b>	<b>Function</b>
-------------	-----------------

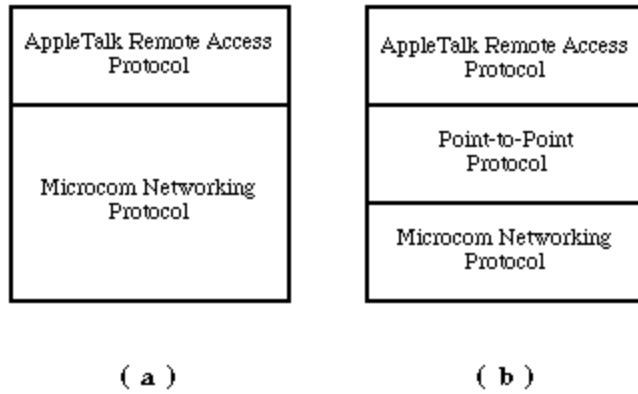
UGNewUser	Creates a new user within the Users & Groups Data File
UGDeleteUser	Deletes a user
UGRenameUser	Renames a user
UGGetUserInfo	Gets information on a user, lists all the users in the Users & Groups Data File, or lists all the users in a group
UGSetUserInfo	Sets password and other data belonging to a user
UGAuthenticateUser	Checks to see whether a given password is correct for the user

### **Calls relating to groups**

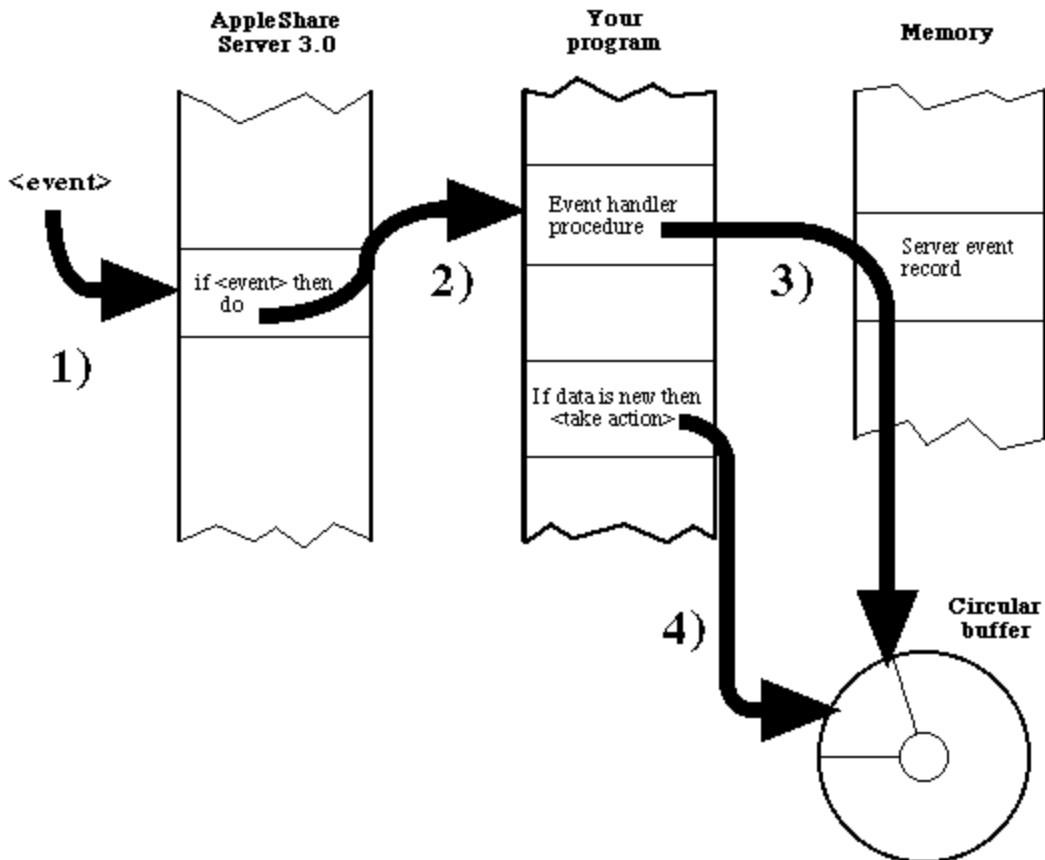
<b>Call</b>	<b>Function</b>
UGNewGroup	Creates a new group in the Users & Groups Data File
UGDeleteGroup	Deletes a group
UGRenameGroup	Renames a group
UGGetGroupInfo	Gets information about a group or lists all the groups in the Users & Groups Data File
UGAssignUserToGroup	Allows a user to become a member of a group
UGDeleteUserFromGroup	Deletes a user from a group

### **Calls relating to the AppleShare File Server or System 7 File Sharing**

<b>Call</b>	<b>Function</b>
UGGetULInfo	Gets information about the AppleShare File Server or System 7 File Sharing
UGSetULInfo (as of December 1991)	Sets the server characteristics



**Figure 1:** The protocol structure for the current AppleTalk Remote Access (a) and the IETF draft specification (b). See text for details.



**Figure 2:** Server event handling allows AppleShare Server 3.0 to execute your code when a certain AppleShare event or event category occurs.

At your program's startup: Your program does housekeeping routines and installs a server event handler (a procedure) that specifies what events will cause your procedure to execute.

1. An event that matches your criteria occurs.
2. At server event time: AppleShare Server 3.0 calls your event-handler procedure
3. At server event time: Your procedure copies the server event record (or whatever parts it needs) into the first empty slot of your buffer.

4. In main-event look: Your program checks the buffer periodically (usually at nullEvent time). When it sees a new buffer entry, it takes whatever action you want it to.

### **Products from APDA**

If you want to work with AppleTalk Remote Access or AppleShare Server 3.0, you may be interested in the following products:

- The AppleTalk Remote Access Developer's Toolkit, APDA part # R0128LL/A—for working with AppleTalk Remote Access.
- The AppleTalk Remote Access Modem Toolkit, part R0129LL/A—for creating CCL scripts that allow a modem to work with AppleTalk Remote Access.

Sometime this quarter, APDA will also make the AppleShare Server 3.0 API Developer's Kit available. This product will be of interest to developers who want to develop AppleShare 3.0-related products.

To order products from APDA, consult the instructions at the bottom of the "Now Available from Apple" column in this issue of Apple Direct.

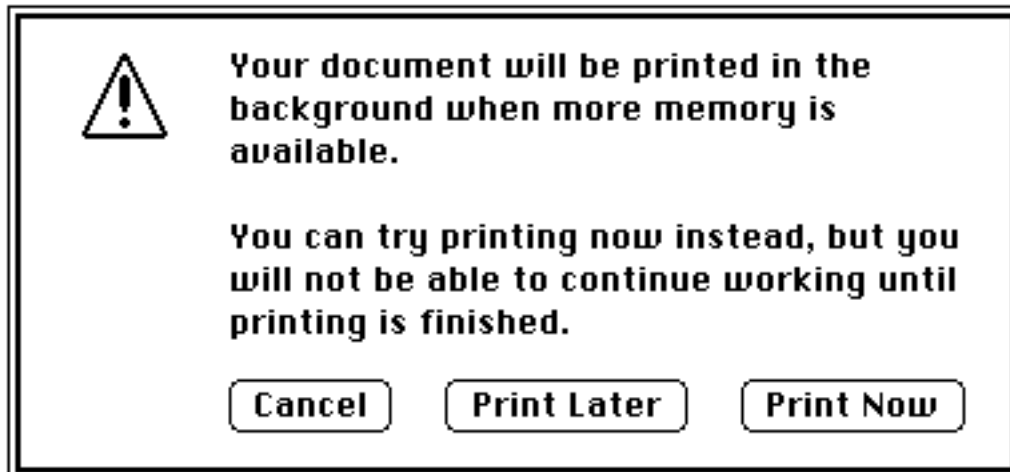


## System 7 Gets a Free Tune-up

At Macworld Expo (and thereafter), Apple is giving every System 7 user a free tune-up—a System 7 Tune-Up disk, that is. While System 7 was successful when we first introduced it, our users gave us valuable feedback and we've responded.

With the tune-up, System 7 makes better use of any Macintosh, including ones in low-memory situations. (And remember that low-memory problems are not exclusively the domain of 2-megabyte machines—an 8-megabyte Macintosh can run out of memory too.) The major enhancements include:

- More-intelligent memory management: This helps every Macintosh use the memory it has and prevents some low-memory dialog boxes from appearing.
- Faster, more reliable printing for everybody: In addition to various internal improvements, the System 7 Tune-Up includes new drivers for the LaserWriter and the StyleWriter. As the screen shot below shows, the new printing software gives users experiencing low-memory problems the option of printing a document in the foreground. (Before, users got a dialog box that told them to quit an application to enable printing.)



- More memory on machines that aren't connected to a network: Users who turn AppleShare off get the use of more memory—an average of 120K.
- Better dialog boxes that help users more: For example, one System 7 dialog box used to tell users, "There is not enough memory to open 'GerbilCalc' (380K needed, 45K available). Closing windows or quitting applications can make more memory available." Now it says, "There is not enough memory available to run 'GerbilCalc'. Do you want to quit the application 'FooWriter,' which has no open windows, and open 'GerbilCalc' instead?"—and your two options are "Cancel" and "Quit Application." Note that the new message is more useful and less likely to confuse the user.
- A faster Chooser: The Chooser now responds more quickly when the user chooses from devices in multiple zones.
- Other small enhancements.

The System 7 Tune-Up disk uses the Installer to make installation easy for users. Apple will distribute the disk free at Macworld Expo and will make it available to users

free through multiple channels: dealers, electronic bulletin boards and information services, user groups, AppleLink, and the Developer CD.

Users will also be able to order the disk direct from Apple for a small fee.

## **Now available from Apple**

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

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

### **Apple Products**

A/UX Developer's Tools for Europe  
B0767LL/A  
\$895.00

A/UX Developer's Tools without MPW for Europe  
B0768LL/A  
\$395.00

A/UX Developer's Tools for International  
B0769LL/A  
\$895.00

A/UX Developer's Tools without MPW for International  
B0770LL/A  
\$395.00

Macintosh Communications Toolbox v.1.1  
M0232LL/F  
\$100.00

Macintosh Programming Fundamentals: Self-Paced Training Course† v.1.0.1  
M0997LL/B  
\$595.00

MIDI Management Tools Set v. 2.0.1  
M0240LL/E  
\$35.00

QuickTime Developer's Kit v.1.0  
R0147LL/A \$195.00

†All customers who have purchased MPF v.1.0 will receive an automatic upgrade to version 1.0.1 at no additional charge.

### **Third-Party Products**

Programming for System 7 (Addison-Wesley)

T0474LL/A

\$26.95

Object-Oriented Technology: A Manager's Guide (Addison-Wesley)

T0361LL/A

\$19.50

## **APDA TOP TEN SELLERS\***

1. E.T.O. Starter Kit & Subscription
2. DAL v. 1.3 MVS/ TSO Server
3. MPW C v. 3.2 bundle
4. MacTCP v. 1.1 License & Developer's Kit
5. Macintosh Programming Fundamentals
6. MPW C & Object Pascal v. 3.2 bundle
7. Inside Macintosh, Vol. I-VI + X-Ref
8. Macintosh Common Lisp v. 2.0b1
9. ResEdit v. 2.1.1
10. AppMaker v. 1.2

\* as of 12/13/91

To place an APDA order from inside the U.S., contact APDA at (800) 282-2732. APDA's number in Canada is (800) 637-0029. And for those who need to call the U.S. APDA office from abroad, the number is (408) 562-3910. If you're outside the U.S., you may prefer to work with your local APDA contact. For a list of non-U.S. APDA contacts, see the "APDA International Contacts and Offices" on page 7 of this issue.

# QuickTime Rolls Out at MacWorld Expo

More than 100 compatible applications shown

At the Macworld Expo earlier this month, Macintosh developers announced more than 100 applications that take advantage of QuickTime, Apple's multimedia system-software extension.

Most of the applications shown were mainstream products, such as word processors and educational programs, that have added QuickTime support so that they can now incorporate dynamic media. WordPerfect, for instance, showed a version of its word-processing program that included video clips, animation, and sound bytes.

Several new classes of products were also introduced, including QuickTime movie editors, CDs with movie clips, and plug-in compression schemes.

To help Apple's field sales force better understand the potential of QuickTime applications, Apple is providing field offices with an electronic version of Apple Intro News, a newspaper released with every major Apple product introduction, which highlights compatible third-party products (both new and upgraded). The paper version of Apple Intro News contains screen shots of the highlighted third-party products, whereas the electronic version instead contains QuickTime movies that demonstrate examples of each product's QuickTime capabilities.

Watch for more QuickTime rollout details in next month's *Apple Direct*, when we'll publish a list of the developers that participated in the Macworld activities.u

## It Shipped

Through the "It Shipped" program, you can announce new and revised third-party products in Apple Direct. It Shipped listings are also made available on the 3rd Party Connection AppleLink bulletin board. You can obtain an It Shipped application by downloading it from the AppleLink network (AppleLink path—Developer Support:Developer Services:Apple Information Resources: Developer Program Information:It Shipped! Program). Or contact Todd Luchette at (408) 974-1241 (voice) or (408) 974-3770 (fax). Once you've completed the application, send it to Engineering Support, Apple Computer, Inc., 20525 Mariani Ave., M/S 42-ES, Cupertino, CA 95014; Attn: It Shipped Program. Or send it via AppleLink to IT.SHIPPED.

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

Publisher	Product (Version)
U.S.A.	
CTA, Inc.	TextPert Developer's ToolKit (1.0)
Chena Software, Inc.	Fair Witness (1.1)
Dantz Development Corporation	DiskFit Pro (1.0)
HeartBeat Software Solutions	MacLife (3.0)
K-12 MicroMedia Publishing, Inc.	Fields of Learning: Life Science (1.00)
King Software Development	Talking Symbols (1.0.4)
Language Systems Corp.	Language Systems FORTRAN (3.0)
LinksWare Corporation	LinksWare (1.0)
MDG Computer Services	QM Log Translator (1.5)
MediaGenic	Shanghai II: Dragon's Eye (1.0)
Michael L. Weasner	Communication Integration Tool (ComIT) (1.5)
Microlytics, Inc.	Spell Finder for use with AppleLink (1.0)
Paragon Concepts	Nisus Arabic (3.21)
Raymond Software, Inc.	MacDiet (1.0)
Roger Wagner Publishing, Inc.	HyperStudio (3.0)
StarTech	RenderEdge
T/Maker Company	ClickArt Newsletter Cartoons (1.0) PowerBundle (1.0) Thought I Could Wallpaper (1.0)
Switzerland	
QuantumSoft	proFit (4.0)

## **Apple Joins Bus. Software Alliance**

At Macworld in San Francisco earlier this month, Apple announced that it is extending support to the antipiracy lobby in Europe and Asia by joining the Business Software Alliance (BSA).

The BSA is an organization set up to protect U.S. software copyrights throughout the world. It sponsors antipiracy education programs, copyright legislation, and piracy litigation.

“Software piracy is a global problem, and joining the BSA is another step toward addressing the business needs of all our developers, both in the U.S. and abroad,” said Kirk Loevner, director of the Apple Developer Group.

“It’s important that Macintosh developers be able to sell their products anywhere in the world without the threat of losing their most valuable asset—their intellectual property.”

The BSA’s mission is to open world trade in legitimate software by advancing strong intellectual-property protection for software. It endeavors to educate users on how software piracy stifles creativity, blocks new-product development, limits users’ options for software solutions, and ultimately results in the loss of billions of dollars to the global economy.

The BSA’s most recent efforts include a push to strengthen Eastern European software-copyright laws and litigation against several Korean and Taiwanese companies accused of piracy.

Other BSA members include Aldus, Ashton-Tate, Autodesk, Lotus Development, Microsoft, WordPerfect, Digital Research, and Novell.

As a member of BSA, Apple now has a voting presence in all BSA activities and a voice in determining how BSA funds will be used. Apple will continue to explore additional ways of partnering with the BSA and others to help fight software piracy.



## CD Highlights, Jan. '92

This column is your guide to the latest Developer Series CD, telling you what's new and notable. To quickly access everything listed below, see the "What's New on This CD" folder (located at Dev.CD Jan 92:Start Here/Read CD License 1st:What's New on This CD?). Here you'll find aliases to every new and updated package on the CD, including all those highlighted below. The January CD, "Winter of Our Disc Content," includes the following highlights:

**Developer Info Assistant and Search Knowledge Base merged!:** The Search Knowledge Base stack has been integrated with the Developer Info Assistant (DIA) stack. To use the search facility formerly found in the Search Knowledge Base stack, click the "Search..." button on the opening card of the Developer Info Assistant.

From the DIA, you will still be able to search the principal technical-information stacks on the CD. These include SplInside Macintosh, Technical Notes, Human Interface Notes, Q & A, Inside Macintosh Vol. VI, Worldwide Development Guide to System Software, Guide to Software Localization, Contents Catalog, and all the develop stacks. For those who can't find what they need on the CD, the Search facility in the Developer Info Assistant allows access to the most-current information in the Developer Technical Answers (DTA) library on AppleLink.

**Sample Code:** We have several intriguing new bits of sample code. AEOBJECT-Edition Sample *f* demonstrates how to use the Edition Manager and the Apple Event Manager as well as some features of the AE Object Model and the AE Object Support Library. MacShell provides a sample framework for writing applications. Kibitz is an updated sample of a network chess application.

Among our January Snippets you will find AECOercionINIT, an INIT that shows you how to install system-level AppleEvent Coercion routines at INIT time. You may also be interested in the coercion routines as samples of AE coercions in general. Also take a look at NamingTableAccess. This Pascal Unit is a small application that demonstrates how to access the NamingTable. It helps dig out the contents of the NamingTable in a TrueType 'sfnt'.

Beware! Items in the Snippets folders are considered Moof!—i.e., "hacks"—and are not guaranteed to be bug-free.

**Networking and Communications:** Look for the new Network Software Installer 1.1. This important contribution is the latest installer package for upgrading AppleTalk, Apple Ethernet, and TokenRing software to support the AppleTalk Remote Access Protocol. Also find the Serial NB Sample Driver 1.0. This is a complete version of the sample driver code provided on the Serial NB Tool disk.

**Testing and Debugging:** Among our new testing and debugging tools, you will find the Super Front End for VU (HITEC). This is a collection of MPW scripts and tools with a menu and commando interface to aid in authoring, executing, and analyzing human-interface tests.

Tests may be written as Virtual User (VU) or MPW scripts. FinderHeaps is another debugging tool; it allows you to track the Finder's heap sizes. The heaps include the Application Heap, System Heap, Code Heap, and Object Heap. You can also ask it to track a fifth heap of your choice.

Also take a look at TV-Man, an application that provides various test-video and -audio patterns, including color bars, gray scale, and concentric-ring patterns.

**International System Software:** We have three new versions of international system software: Czech 6.0.7, Hungarian 6.0.7, and Turkish 7.0.1.

**Other Exciting Updates:** Among our updates, notice the latest version of AppleGlot, our localization tool for the Mac. Both MacsBug (6.2.2.) and ResEdit (2.1.1) have been updated to the latest rev. Also see the new StyleWriter 7.2 software, which was released just this month. This new driver for the StyleWriter allows you to print at up to twice the print speed the previous StyleWriter drivers allowed. And last but not least, we have the new software for the Apple Macintosh Display Card 8•24 gc.

To quickly find all our new and updated software, go to the "What's new on this CD?" folder in the Start Here folder of your January CD.

## **APDA International Contacts & Offices**

APDA products can be purchased through several international programs. Contact the international APDA programs listed below if you are interested in purchasing tools and documentation locally.

### **Australia**

Charmaine O'Keefe  
AAPDA  
16 Rodborough Road  
Frenchs Forest NSW 2086  
Australia  
AppleLink: AUST0214  
Telephone: 61-2-452-8245

### **Austria**

Rainer Bernert  
Apple Computer Ges.m.b.H  
Ungargasse 59  
A-1030 Vienna  
Austria  
AppleLink: BERNERT.R  
Telephone: 43-1-711.82-27

### **Czechoslovakia**

James Guidi  
TIS  
Evropská 94  
16000 Prague 6  
Czechoslovakia  
AppleLink: CZ.IMC  
Telephone: 42-2-311-92-84

### **Denmark**

Ole Stakemann  
APDA Nordic  
Fortunvej 55  
DK-2920 Charlottenlund  
Denmark  
AppleLink: DK0016  
Telephone: 45 31 64 10 68

### **Egypt**

Tarek Thabet  
PACC  
70, Gameat El Dowal  
El Arabia Mohandessin

Cairo  
Egypt  
AppleLink: AMME.COMM  
Telephone: 202-3-481.381

**Finland**

Ole Stakemann  
APDA Nordic  
Fortunvej 55  
DK-2920 Charlottenlund  
Denmark  
AppleLink: DK0016  
Telephone: 45 31 64 10 68

**France**

Marcel Averbuch  
Prim'Vert  
36 Rue des Etats-Generaux  
Versailles 78000  
France  
AppleLink: PRIMVERT  
Telephone: 33-1-39-02-33-44

**Germany**

Rupert Holzbauer  
Apple Computer GmbH  
Gutenbergstrasse 1  
D-8045 Ismaning  
Germany  
AppleLink: GER.APDA  
Telephone: 49-89-99640-176

**Greece**

George Georgaras  
Rainbow Computer Applications S.A.  
El. Venizelou 184  
Kallithea, Athens  
Greece  
AppleLink: IT0026  
Telephone: 30-1-959.40.82

**Hungary**

Andras Szigeti  
Graphisoft Trading Co.  
1146 Budapest  
Hermina út 35  
Hungary

AppleLink: HUNGARY.IMC  
Telephone: 36-1-121-66-93

**Iceland**

Ole Stakemann  
APDA Nordic  
Fortunvej 55  
DK-2920 Charlottenlund  
Denmark  
AppleLink: DK0016  
Telephone: 45 31 64 10 68

**Israel**

Igal Dekel  
Yeda Computers Ltd.  
110 Igal Alon St.  
Tel Aviv 67891  
Israel  
AppleLink: IT0029  
Telephone: 972-3-330-743

**Italy**

Paola Reitano  
ESSAI srl  
Via Cesare Correnti 15  
20123 Milano  
Italy  
AppleLink: ITA.APDA  
Telephone: 39-2-27326334

**Japan**

Taiju Tachibana  
MANPOWER Co., Ltd.  
No. 2 Anzen Bldg.  
6-6 Moto-Akasaka 1-Chrome Minato-ku, Tokyo 107  
Japan  
AppleLink: APDA.JAPAN  
Telephone: 81-3-3478-6531

**Norway**

Ole Stakemann  
APDA Nordic  
Fortunvej 55  
DK-2920 Charlottenlund  
Denmark  
AppleLink: DK0016  
Telephone: 45 31 64 10 68

**Poland**

Bogdan Jedrzejczyk  
S.A.D. Ltd.  
u1. Mangalia 4  
02-758 Warsaw, Poland  
AppleLink: POLAND.IMC  
Telephone: 48-2-635-54-97

**Portugal**

Paula Gentil-Homen  
Interlog, Informatica, SARL  
R. Prof. Mira Fernandes, Lote 20/21 r/c  
P1900 Lisboa  
Portugal  
AppleLink: IT0040  
Telephone: 351-1-8470513/20

**Saudi Arabia**

Dr. Abdullah Al-Hamdan  
Jeraisy Computer Services  
Makkah Road (Islam Road)  
P.O. Box 317, Riyadh 11411  
Saudi Arabia  
AppleLink: SAUDI.IMC  
Telephone: 01 4624597

**Spain**

Juan Carlos Rubio  
APDA Spain, SYMSA  
C/ Doctor Fleming 54  
1 Dcha., 28046 Madrid  
Spain  
AppleLink: SPA0036  
Telephone: 34.1.2506618

**Sweden**

Ole Stakemann  
APDA Nordic  
Fortunvej 55  
DK-2920 Charlottenlund  
Denmark  
AppleLink: DK0016  
Telephone: 45 31 64 10 68

**Switzerland**

Urs Binder

Industrade AG, APDA  
Apple Computer Division  
Hertistrasse 31  
CH-8304 Wallisellen, Zürich  
Switzerland  
AppleLink: URS  
Telephone: 41-1-832-8276

**Turkey**

Yakup Dursen  
Bilkom AS  
Abdi Ipecki Cad 16/3  
80200 Nisantasi  
Istanbul UB11 1BB  
Turkey  
AppleLink: IT0112  
Telephone: 90-11-32-15-06

## **Developer Resources**

Get answers to many of your development questions, expand your knowledge, or find help for your programming projects by tapping into these resources.

Developer Associations and User Group Programming Special Interest Groups (SIGs) These groups can provide programming resources such as informational meetings, bulletin board services, technical tips and techniques, and contact with other developers to help with your development projects.

- User Groups (general). To locate an Apple User Group Programming SIG near you in the U.S., call (800)538-9696, x500.
- MacApp Developers Association. Dedicated to supporting users of MacApp. For membership information, call (206) 252-6946.
- SPLASH Developers Association. Dedicated to supporting users of Symantec programming languages. For membership information, call (415) 527-0122.
- Berkeley Mac User Group Programming SIG. For membership and benefits information, call (415) 849-9114.
- Boston Computer Society Mac Tech Group Programming. For membership and benefits information, call (617) 625-7080.

## **Macintosh Development Services Directory**

This directory of third-party development-related service providers can help supplement the needs of your development efforts. Experts are available covering the entire development spectrum including technical trainers, contract programmers, product localizers, product testers, and technical documentation writers. To order, contact APDA (see below).

## **Important Apple Phone Numbers**

- Ordering APDA products. (800) 282-2732 (U.S.); (800) 637-0029 (Canada); (408) 562-3971 (other countries). For other local non-U.S. providers, see a complete listing elsewhere on this page.
- Apple Software Licensing. For information on licensing Apple software for site, internal, or commercial distribution, call (408) 974-4667.
- Developer Support. U.S. and Canadian providers of commercial products and services can find out about Apple's developer support programs by calling (408) 974-4897. For non-U.S. support programs, see a complete listing in the latest APDA Tools Catalog.
- Developer Training. Apple Developer University offers course and self-paced training products for aspiring and advanced programmers. To obtain a course catalog or register, call (408) 974-6215.
- Apple Consultant Relations Program. Write us and find out how you can build solid relationships with the professional computer systems consulting community throughout the U.S. Address: Apple Computer, Inc., Apple Consultant Relations, 20525 Mariani Avenue, M/S 36AJ, Cupertino, California USA 95014
- U.S. Apple Authorized Dealers. To locate a U.S. dealer near you, call (800) 538-9696.



**Authorized Third-Party Developer University Training Sites**

Selected Developer University courseware is available through regularly scheduled classes and, in some cases, on an on-site basis at the following training locations.

- EDS, 5228 Tennyson Parkway, Plano, TX 75024 (214) 403-5261.
- University of Maryland, Computer Science Center, College Park, MD 20742-2411 (301) 405-2956.
- Cornell University, Cornell Information Technologies, 220 CCC Garden Avenue, Ithaca, New York 14853 (607) 255-4983.
- University of Oregon, Portland Center, 720 S.W. Second Avenue, Portland, Oregon 97204 (503) 725-3055.

Please contact these offices directly for information about classes, locations and fees.

## **First the Column, Now the Book**

Over the years, we've received hundreds of requests that we turn Tog's columns into a book so that his readers might have a lasting source of his great and abiding wisdom. Up until recently, we turned these requests down, encouraging Tog to spend more time on his column and less time writing these requests.

In the last year or so, however, people other than Tog began making the same requests. At first, we thought that Tog was calling in favors from his friends, but we discovered that real, bona fide Apple Direct readers out there have actually been demanding a more permanent source of Tog. And now you've got it: all the Tog you'll ever want, straight from the horse's mouth.—Editor

After three long years of work, you and I have written ourselves a book, one chapter—column—at a time. I decided in late 1988 that I wanted to write a book on interactive software, but I wanted to make the book itself interactive so it would not end up being dull, dry, and nonresponsive. Fortunately, you folks were willing to chip in, with letters, Links, and all kinds of moral support.

Now, we've done it: Tog on Interface, from Addison-Wesley is 300 pages of questions, answers, responses, retorts, and endless essays available at your neighborhood purveyor of fine literature right now.

(You might be assuming that, characteristically, I named the book to grab all the credit for myself. Nothing could be further from the truth. I wanted to name it something far less personal, like A Popular Human-Interface Evangelist Speaks Out on Interface, but Addison-Wesley insisted on naming it after me. I fought them tooth and nail, but in the end, they prevailed.)

More than half of the book has never appeared in Apple Direct: I've fleshed out the original writing with updated material; unpublished letters from many readers; new essays; and, by popular demand, a liberal sprinkling of actual scientific references. Below are some excerpts from the book.

### **In the Beginning...**

In the early days, developers wishing to break into the marvelous new world of the Macintosh were forced to make pilgrimages to Cupertino to sit at the feet of the masters (an unpleasant prospect at best) in the hopes of being able to capture the essential flame of the Macintosh philosophy. But after the Macintosh shipped, the developer community began to swell to ever larger numbers and started fanning out across North America and around the world. I soon began seeing programs that followed all the specifics of the Guidelines but were clearly not "Macintosh." These programs were coming from developers who had had no direct contact with the Macintosh team. Something intangible but vital seemed to be lost when people only had the Guidelines.

The Macintosh interface was more than a cluster of specific guidelines; it was built upon principles, a rich set of rules which were embodied in a culture passed down from one person to another. There was no written history of the principles that drove the Macintosh. In fact, there was not even an oral history: None of us could even put into words what this interface was about.

The principles were so ingrained in the culture that they were taken as natural laws. After all, one need not write down gravity: The next guy is bound to catch on

when he finds himself unable to fly. The principles were felt, rather than spoken of, and understanding them required immersion in the Macintosh culture.

It became clear to me, as the success of the Macintosh grew, that what had worked in the early days to promulgate the culture would work no longer. I launched a research project, headed up by Kristina Hooper, to identify and codify the principles upon which the interface is built, and we set out to capture the elusive spirit of the Macintosh.

Those first principles helped provide a foundation, but there was still something missing, and aberrant applications continued to appear. It was then that I decided to begin writing this book. I wanted to be responsive to the real needs of real designers, so I moved from research into the Apple Evangelism Group, where I can interact every day with new developers and new projects. But even that wasn't enough: I wanted the book itself to be interactive. Human-to-human interactive. So I started a monthly question-and-answer column in Apple Direct.

Now, three years later, the book is done...The issues discussed are real and relevant, having arisen from letters sent to me at Apple and from visits I was making to developers around the United States and around the world.

The columns, and thus the book, were written against the backdrop of System 7, the development of which was under way at the time. The secrecy surrounding the project made it impossible for me to be as candid about our plans for System 7 as I might have liked: As you read through the book, you will find me periodically discussing changes that "may be made someday in the far distant future," changes that, in fact, were already under way.

The column also was a device for gathering "public opinion" from Apple's developer community, and many of the changes in System 7 that eventually did occur were the result of or strongly influenced by letters, both those published and not published, sent in to the column.

### **The Visible Interface**

Problems are arising in visible user-interface applications today, problems for the first time making it difficult for people to understand and use them.

I have purposely used the term "visible interface," rather than "graphical interface," because there is a difference: A visible interface is a complete environment in which users can work comfortably, always aware of where they are, where they are going, and what objects are available to them along the way.

To be labeled a graphical interface, an interface need only make use of objects that have a distinct graphical representation. Many aspects of the graphical interface may remain invisible.

A library, with its visibly structured and labeled collections, oaken card catalogs, librarians, and lots of light, has a visible interface.

The library's interface is also a graphical one, in that the library is populated with objects having unique graphical representations, such as books, "Shhhhhh" signs, librarians, and fire extinguishers.

These objects can be easily differentiated by their unique appearances (although the woman who ran the library in my elementary school did bear an uncanny resemblance to a fire extinguisher).

In contrast to the library, consider yourself driving late one moonless night on unfamiliar country roads. You see plenty of graphic objects in the form of signs, warning you of deer in the area, alerting you to falling rock, telling you you can shed pounds and inches by drinking lots of light beer. Inevitably, you will run across a familiar, eight-sided red icon informing you that you must stop, even though there is not a soul within 500 miles. (If you fail to stop, of course, you will see a bright red flashing light signifying that there was one person within 500 miles.) There are many graphical objects in the interface of that dark country road, but the actual navigation is invisible, and you may very well become lost.

The Tunnel of Love, with its day-glow monsters (graphically represented objects) lining a pitch-black tunnel might, at first glance, also seem to be a graphical user interface, even though users of the Tunnel of Love never have any idea where they are and where they are going. (Of course, given a friendly traveling companion, they don't really care.) But the Tunnel of Love is not really a graphical user interface at all; it is an interface for displaying graphics. The monsters are not your friends.

There is no intrinsic connection between graphics as the subject matter of an application—even when those graphics appear as distinct objects—and graphics as a driving element of the interface itself. If the Tunnel of Love's many iconic signs, such as the familiar red stop sign that displays the warning "Stop! Bottomless pit straight ahead!" actually were both truthful and helpful, the Tunnel of Love would be magically transformed into a graphical user interface, but so much of the rest of the interface would continue to be shrouded in darkness, it could hardly be termed a visible interface.

Interfaces resembling the Tunnel of Love are based on a metaphor called the black cave metaphor. These interfaces are left over from an earlier era, an era in which graphics played no part at all.

### **From the Black Cave to the Light of Day**

When I first became involved with computers, back in 1958, the interface typically consisted of flashing lights, punched cards, and TeleType printers.

Eventually the original TeleType printer was replaced with a computer monitor, but the interface designers dutifully reproduced the actions of the historic printer in the green light of the new screens, faithfully duplicating every limitation of the mechanical device in the process.

Then came the visible user interface, with its rich use of graphics, consistent behavior, visually apparent structure, and clear communication. It rose from a culture that started at Stanford Research Institute and spread throughout Silicon Valley, a culture dedicated to the single task of bringing the power of the computer to people everywhere, instead of concentrating it among a select priesthood.

This primordial graphical user-interface culture eventually produced the first visible, graphical interfaces, built on metaphors based in the real world, interfaces such as the Xerox Star, Lisa, and Macintosh. Non-computer professionals for the first time gained a sense of competence and control over the computer. They knew where they were and what they were doing at all times. People were no longer lost and confused.

Recently, some applications and entire computer interface systems based on graphical user interfaces have begun to lose the visibility of their underlying structures. These semivisible graphical user interfaces are found in systems that are filled with

lots of pretty windows and all sorts of check boxes and buttons but have left key functionality hidden and invisible.

Such lack of visibility is particularly bewildering to users of a graphical user interface, because it seems as though everything really is visible—after all, there is all that graphical stuff on the screen.

If one continually collides with the walls of a black cave, one accepts that one cannot see well in pitch darkness. But should the same thing happen in a clean, well-lit library, one can only dwell on the possibility of a brain tumor, or worse.

Programmers can use any kind of interface. They have exceptional minds that embrace, enjoy, even thrive on abstract, invisible interfaces. They assume that everyone else has a wonderful memory and enjoys manipulating abstract symbols. (Although many people may have good memories, I think the current dearth of Algebra Fun Clubs And Family Centers in our communities speaks volumes about our collective delight in manipulating abstract symbols.)

But people want to operate in a self-consistent, stable, visually apparent virtual reality, populated with visually apparent objects that, when manipulated, will behave in predictable ways. They need to be able to see the structure of the world in which they are operating, and they need to explore without fear of becoming lost or of causing irreversible damage.

The Macintosh interface is one example of such an interface. It offers a visual/behavioral “language” that is simple, clear, and consistent. Users interact with the system in a way that closely mimics the way they interact with the real world.

On the original electronic computers, there was no metaphor between the user and the raw reality of vacuum tubes and wires. In today’s visible interfaces, that raw reality has been replaced with a softer, virtual reality—an illusion spun of nothing more than light and logic.

Want to read more? Look for *Tog On Interface*, from Addison-Wesley, in a bookstore near you.

## **GetNextEvent**

All of the following trade shows/events find Apple Computer, Inc., scheduled to exhibit, as of press time. This list may be incomplete. If you have information about a show you want listed here, write to Apple Direct Event Calendar, 20525 Mariani Avenue, Mail Stop 75-2B, Cupertino, CA 95014. For further information, check the Events folder on AppleLink (path—Developer Support:Developer Services:Apple Information Resources: Developer Events).

February 12 through 14  
GTC Southwest, Austin, TX  
Contact: Yubi Wahlquist  
(916) 452-4902

February 19 through 21  
Seybold, Boston, MA  
Contact: Kathleen Kaiser  
(213) 457-5850

February 20 through 22  
Macworld Expo/Tokyo, Makuhari, Chiba, Japan  
Contact: Junichi Kawaminami  
(03) 5562-6230  
Fax: (03) 5562-6060

February 23  
Education and Multimedia Symposium, Tokyo, Japan  
Contact: Masashi Okumura  
(03) 5562-6250  
Fax: (03) 5562-6060  
February 27 through 28  
Multimedia Los Angeles, CA  
Contact: Mitch Hall & Assoc.  
(617) 361-8000

March 9 through 12  
NCGA—National Computer Graphics Assn., Anaheim, CA  
Contact: Martha Filson  
(703) 698-9600

March 10 through 12  
CD-ROM Conference and Expo, San Francisco, CA  
Contact: Cahners Exposition  
(203) 964-0000

March 18 through 21

SPA, Seattle, WA  
Contact: SPA Conference Dept.  
(202) 452-1600

## Ten Common Product Launch Mistakes

by Leigh Marriner, Marriner Associates

Sometimes, especially in small software companies that are resource-strapped, you're launching your first or second product at a time when most people in the company are almost too busy just getting the product out the door to think about marketing. You're pushed for time, your resources are strained to the max, and THE DEADLINE makes all else seem unimportant. So it's no surprise that we've all made some of the most common launch mistakes—and sometimes more than once.

There's a lot of time and money riding on the launch of your product and a lot to be gained by doing it the best way possible. Here's a checklist of the ten most common product launch mistakes and how to avoid them.

**#1: Shipping a second-rate product.** The best marketing campaign can't make up for a lackluster product. Product quality is still the key to success in this industry, although many good products die for lack of adequate marketing support.

Launching a weak product is usually a waste of money. This kind of product tends to struggle along; a small group of committed users continues to buy it, but it never gets a mass audience. The developer never makes a profit, because the costs of keeping the product alive outweigh the revenue generated.

To avoid this, once you're nearing the end of product development, you need to ask yourself some very tough questions and be (sometimes painfully) candid in your answers: "Is the product really worth publishing?" "Are customers willing to pay for this product?"

The investment of several man-years in developing the product isn't enough to justify bringing it to market. A product that started out as a brilliant idea may have been implemented poorly. Or the competitive situation may have changed since development started. To make the product marketable, you may have to invest several times your development costs in marketing, sales, tech support, and manufacturing.

Don't let the already-spent development dollars keep you from being prepared to consider the option of abandoning the product. Deciding to abandon your "dream" product is certainly one of the most difficult choices you'll ever have to make, but weighed against the possible alternative of bombing in the marketplace, it may be the wiser course of action.

Too often, companies are blind to their product's potential—or lack of it—by the time it gets to the end of the development cycle. If you have questions about whether the product is good enough to ship, try letting 10 to 20 target customers use it and then listen to what they have to say.

Gauge their enthusiasm. Do they understand why it is so good? Is the product advantage important to them? If they don't rate it an 8 or higher on a 10-point scale, you're in trouble.

If you do launch the product, pay attention to the early reaction from the press, industry influencers, bulletin-boards users, and so forth. If there isn't a strong positive reaction, then you may want to reconsider making any further marketing investment unless you also make substantial product changes.



**#2: Shipping a marginally better product (corollary to Mistake #1).** Shipping a me-too product in an established category, without a strategy for overtaking the category leader, doesn't work. Being 50 percent better than the category leader isn't enough. Unless the product is at least 100 percent better, the product differences alone aren't enough to make the necessary impact on the market. (And it's rare to see products that really do leapfrog the leader.)

Almost all the profit in a product category is made by the Number 1 and Number 2 players. Being Number 3 or 4 may generate revenue but not much profit. Don't let the size of a category fool you into thinking there's easy money to be made. It's better to be top dog in a smaller market segment than one of the pack in a large one.

To successfully launch a product in an established category, you need to define a reasonably-sized and growing segment in which your product can become the leader. Identify the circumstances in which your product offers users real advantages. Pick a segment in which you know how to reach the customer and in which you can target your limited marketing resources. Segments can be defined by hardware platform, specific vertical markets (such as magazine publishers), demographic group (such as K-5 children), channel, and so forth.

**#3: Ignoring feedback from users.** Alpha and beta tests are critical for finding bugs and also for gauging how well the product will be received by users. Too often, beta tests aren't done at all or aren't done until it's too late to act on the feedback. Developers often feel that there isn't enough time in the development cycle to allow three to four months for a beta test.

Even if your company has an excellent internal testing department, real-world beta tests are invaluable for finding unexpected anomalies when the product is used with various hardware configurations. If you fail to get this kind of feedback or don't give it its due, it can result in the kind of mistake from which a smaller company can't easily recover.

Measuring user acceptance at the beta stage is often even more important than testing for bugs. For example, I was involved in launching a product that helped users collate group writing projects. If we had really listened to the feedback, we might have realized sooner that the people who stood to benefit the most from the product were low-level collators—and they weren't usually in a position to recommend the purchase. In contrast, by listening to new users talk about their experiences in getting started with a new adventure game, we learned that it took too long to build a new playing character. So we decided to ship the product with premade characters. This is truly the point at which to ask the sometimes politically difficult questions about whether a product is viable and ready to ship.

**#4: Waiting too long to plan the marketing program.** How often have you finished a product, heaved a sigh of relief, and then said, "Now we have to hire a marketing expert and launch the product"? A marketing plan should be completed and ready to implement at least four to six months before the ship date.

Doing so will enable you to save money in many ways. Money won't be wasted on rush charges due to the innumerable last-minute changes caused by starting late. You'll have time to undertake pieces you can do yourself, such as writing first drafts of

box copy or the press release, without having to pay someone else to do it to meet an almost impossible deadline.

Most importantly, developing a marketing plan far in advance lets you have an overall view of the objectives for the product and how you are going to accomplish them with your scarce resources—before you start spending money.

Otherwise, commitments for longer lead-time items such as ads, direct-mail campaigns, and reseller promotions will be made before you know whether they are critical to your launch. Too often you may decide that you must create a presence at Comdex or Macworld, spend \$15,000 before all is said and done, and then realize that the trade show was only a third-level priority—and that the money spent is badly needed elsewhere.

There is no cookie-cutter approach to deciding what the elements of a marketing plan should be. The key is to ask the following kinds of questions to determine what marketing approaches will work best for your product, and then build the answers and resulting required actions into a plan:

- Is this an emerging market in which it is hard to identify scattered prospects, which necessitates your investing in lead generation? Or is this a mature market in which making the purchase process easy is most important?
- Exactly who are the customers?
- Where can you reach the customers?
- Where do customers get information? Who are the influencers?
- How can you overcome customers' reservations to get them interested?
- Is a trial-use or demonstration program necessary?
- What is the customers' purchase process? Who has to approve the purchase?
- How can you make the purchase easy?
- How much sales support is required?

Answering these questions and others like them will keep you from undertaking programs without thinking through the objectives and challenges for your specific situation.

**#5: Failing to draft a product-direction sheet.** Before you start producing any materials and writing copy, it helps to develop a direction sheet to guide the marketing staff and creative agencies regarding what should be said about the product. This guide should specify the most important advantages and benefits of the product so that all staff, consultants, and agencies are working from the same script.

Package copy, press releases, sales promotion items, advertising, direct mail copy, and all other customer communications should be based on this document. At the very least, the direction sheet should include the following:

- A one- or two-sentence description that precisely and concisely states what the product does and the target audience.
- The one unique selling point. What makes this product different and better?
- Secondary consumer messages.
- Dealer messages.
- Key features and benefits.
- The typical applications the product is used for, or examples of game play.
- Price.
- The competitors.

Drafting a direction sheet early in the launch process has several advantages. It forces you to think through what is really important about your product. It also helps guarantee that all the marketing materials will work together and that you don't send mixed messages to customers (for example, it helps ensure that you won't have different tag lines on the box and the product's sell sheet).

Customers need repeated exposure to the same message to be motivated to buy, and all materials should work together to convey a consistent message.

It also prevents the art department from driving the creative process. The marketing and communications aspects should be paramount. For example, we've all seen pretty packages that would make nice posters, but you can't read the product name on them, or the tag line isn't descriptive. Using a direction sheet ensures that everyone is aware of the communications objectives.

Also, having the product's direction on paper means that you won't waste time rethinking the features and benefits of your product each time a new communications piece has to be written. Your consultants and agencies will work more efficiently, which will save you money.

**#6: Failing to focus on a single distinction from the competition.** Every software product has competition, even if it is just the nonelectronic way of doing something. Most customers are going to absorb only one major point about your product. There has to be one crucial factor that compels them to take the time to try it rather than continue to do things as before. Customers are liable to ignore a complicated message. A laundry list of features doesn't pack the same punch as information about one easily understandable advantage.

Ideally, you build a product that you can describe in one sentence so the intended user will respond. Building a product that delivers a specific benefit is the most cost-effective way to communicate with customers. Smaller developers often don't have enough marketing dollars to effectively convey a more complicated message.

**#7: Planning the ads to begin on the product's ship date.** As we all know, most products aren't ready to ship on their original ship dates. Almost always there are last-minute delays: The manual needs to be rewritten, the pre-holiday rush makes the printing company take longer than planned, the last bugs take longer to fix than anyone expected, and so on and so forth. Since advertising and many other marketing programs have to be planned four or more months in advance (for adequate production and space reservation time), it is often difficult to decide exactly when to start running them. There is usually pressure to have the ad campaign premiere as soon as possible, to begin building demand for the product.

However, a good rule of thumb is to never plan to run an ad until at least one month after the stipulated product ship date. If your ad runs before the product is available, you haven't gotten the most out of your money.

It takes one to three weeks for initial stock to make it into the channel. The customer who reads the ad but can't find the product anywhere will have forgotten about it by the time the product ships. This rule of thumb is especially true for small launch budgets, where "teaser" or "preview" ads aren't feasible.

**#8: Relying on a big ad campaign to move the product.** Relying on magazine ads alone is a lazy approach to marketing. To many people, marketing equals advertising. However, a business product usually requires a minimum advertising expenditure of \$100,000 before the ads start to build significant product awareness. This is just too expensive for small developers (unless you create a direct-response ad that pays for itself).

There are options that are more prudent and valuable. You may find it more worthwhile to invest in better packaging or to put your CEO on the road to talk to the press, influencers, user groups, industry consultants, and so forth.

**#9: Being stingy with evaluation copies.** Building word of mouth and influencing the inner circle is the most important factor in the success of new products. Fully functional evaluation copies are one of the least expensive advertising vehicles. Give them to the press, dealers, store salespeople, industry gurus, user groups—anyone who talks to others about software products.

Giving dealer sales representatives a free copy of your product can be an especially powerful method of influencing them to recommend it. There are only two ways to get salespeople's attention: Have customers ask for the product, or have the salespeople be users themselves.

In a survey we did for an entertainment publisher, we sent copies of a sports game to the salesperson specializing in games in 500 stores. Four months later, we measured the effects and found that 90 percent of those who had received a copy had played it, versus only 60 percent for the leading game in that category. And most of them had talked to other sales staff about the game and felt that it helped them answer customer questions.

**#10: Expecting dealers to push the product.** This mistake should be old hat by now, but I still talk to developers who think that if they can just get their product onto a dealer's shelf, their worries are over. In most reseller chains, a product has 6 to 12 weeks to prove itself and move the required number of copies per store. If it doesn't, it is off the shelf permanently. No marketing campaign can build sufficient demand in this time period if it isn't already in place and working when the product hits dealers' shelves.

Dealers put the product onto the shelf, but they expect you to stimulate the consumer demand to move it out the door. Dealers carry thousands of products, and in most cases the salesperson tries to sell whatever is easiest—either what the customer asks for or what is already moving well. Resellers do provide several (often expensive) marketing programs and are willing to work with you to create demand. But many small developers cannot afford to spend \$60,000 to \$80,000 working with one chain on a program that will last for less than a month.

Much of the product launch process is just common sense, yet at one time or another, veterans and novices alike make the mistakes discussed here. The next time you see a new product coming down the pike, resensitize yourself to the potential dilemmas; take a few minutes to rethink these all-too-common mistakes—even if you've done many launches. Odds are that it will be a few minutes well invested.

Happy launching.

Leigh Marriner is the founder of Marriner Associates, a firm located in San Rafael, California that specializes in developing competitive strategies and marketing programs for soft- ware companies.

## **The Advantages of a Site License Program**

### **Why We Offer It**

Although there are certainly many pros and cons to the complex issue of site licensing, this marketing technique has worked well for us. On Technology has found many situations in which the benefits are well worth the time and effort involved to institute a site-license program. The bottom line is that our larger customers need alternative ways to purchase mass amounts of products.

For us, the secret to making the site license an effective sales agreement has been listening to what customers want, understanding why they want it, and then creating a win-win agreement that meets the needs of both parties (see "Why Customers Ask for Site Licenses," on page 12). We've found that this has often meant the difference between getting the account or getting the boot.

It's never easy for a company to decide whether it should begin offering site licenses. We made the decision based on the type of products we sell, workgroup applications that run on networks.

We felt that our products' success depended on our ability to get them accepted and utilized on every node within a large network, and that dealing directly with larger customers was the most effective way to achieve that. I hope that this article will give you some ideas and define some issues to think about to help you make the right decision for your company.

I can almost guarantee that if you decide to implement a site license program, you won't be 100-percent successful the first time out. It will take some trial and error, but don't get overwhelmed or totally frustrated.

Just make sure that you think about the details not only from the customers' perspective but also from your own internal viewpoint. Define the internal systems, processes, and procedures you'll need to effectively administer and maintain a site license program.

In general, whether site licenses are right for you depends on the kinds of products you offer, how flexible you can be in your distribution arrangements, and how well you understand customers' needs. Here are some of the things we've learned through our experiences.

### **Defining "Site License"**

Large corporate customers have been asking for site license programs for years, but when you're sitting across the table from them, the first thing that's important to discern is what they actually mean when they use the term site license. It has become a catchall phrase for many different implementations.

For example, some large customers have instituted a centralized electronic network to distribute software to users. Their idea of a site license usually is to receive "Golden Disk" privileges, in which a master copy of an application is put onto a server. All the users can then download the application for use on their machines. Some customers push to pay one price for unlimited enterprise-wide use; others pay for a specified number of users, with the stipulation that they'll pay extra if the total usage exceeds the agreed-upon commitment.

Regardless of a company's purchasing or administrative structure, it reaps a cost benefit by committing to a certain volume. Often, what customers really want when they

ask for a site license is a Volume Purchase Agreement (VPA). Corporate customers feel that if they are willing to commit to hundreds, or thousands, of copies of your product—and to take them all in one order—there should be some price compromise on your part.

This typically means offering a better price for shrink-wrapped software than even the most aggressive reseller can offer at the same volume. This option serves many companies very well, because it meets their needs whether their purchasing process is highly centralized or very decentralized.

The basic difference between a site license agreement and a VPA is really the way the software is delivered. With a VPA, products are usually delivered in caseloads or in special multiuser packages with limited documentation.

We've found the VPA negotiation to be somewhat easier, because the license and warranty information is already in the package; there is less need for the legal departments to get heavily involved. Most of our agreements with large corporate customers tend to be the VPA kind.

### **Products That Lend Themselves to Site Licenses**

Site licenses aren't ideal for all kinds of products. Don't waste precious sales resources trying to sell site licenses if your product is an unlikely candidate. There are two basic criteria that will help you determine whether your product may be well suited for a site license or a VPA: product category and competitive position.

Certain categories lend themselves to these arrangements because of how the products are used. Often, if a user organization must standardize on the product for it to work effectively, you have a candidate. Network-based products such as our Meeting Maker product come to mind. It is ideal to license because, much like E-mail, it will be installed on every machine in a department or organization. All users on a network need a copy for the product to work effectively, so it is a good candidate.

Also, if you develop an application in a new product category or one that fills a specific need that currently isn't being addressed, there seems to be more urgency on the part of corporate managers to get it installed in the entire user base. Utilities in a new category are particularly good candidates in this respect.

On the other hand, products in existing application categories such as spreadsheet programs or word processors are usually more difficult to site-license, because customers have already acquired many of these after many years of purchasing. Also, categories such as graphics packages or databases tend to be unattractive candidates, because they typically are not put on every user's machine; they are highly specialized apps that are used by fewer people.

The other determinant is your product's competitive position within its category. If you're up against one or two strong market leaders in a particular category and the fight for market share is tough, it may make sense to explore site licensing. The idea is that there are a certain number of large companies that tend to be much more concerned about price than anything else when making a purchase decision.

A developer who knows how to successfully target such accounts can make some strong market-share inroads with an aggressively priced site-license or VPA program.

### **Developer Benefits**

I believe that if done correctly and wisely, site licensing can have many benefits for developers. The first is competitive entrenchment. We all know that in the Macintosh market, the MacWEEK 200 represent well over 50 percent of the installed base of machines. This "hit list" of 200 companies is easier to reach if you use an effective sales strategy.

If your product can become the standard in its category in even 10 percent of these accounts, you've established yourself as a market force and have begun to build barriers for your competitors.

Developing a site license or VPA program that addresses customer needs and that is profitable for the developer will not only help prevent competitors from establishing a foothold in these companies, but will also help build a solid reputation that can smooth the way for you to other companies. If you're known to be "hot" with large-company X, then your reputation will precede you to the next level of accounts; your selling job there will be easier.

Another benefit is high revenue in a relatively short sales cycle, which has been particularly important to On Technology. Although the selling cycle necessary for closing a site license or VPA deal is still a long one, overall I think that for the amount of revenue it produces, the cycle is shorter. In effect, in one order, you can get what might otherwise take your sales rep a year, and many customer visits, to accomplish. This is particularly important, because most Mac-only developers are still trying to build a stronger revenue position. We've found that sacrificing some money on the cost-per-user side of the equation in exchange for more paid-up-front users is usually a winning situation.

The one-time large purchase also tends to increase your longer-term revenue by generating more upgrade potential. These days it seems as if most developers are releasing upgrades about every 12 months. A site license program can increase the number of upgrades sold.

So if your company is still somewhat small (less than \$10 million), I believe that it makes sense to go for the larger up-front order. As your product line matures and your business grows, there will always be the opportunity to increase your revenue per user, but to help you get successfully through the short-term growth spurt, I think that the overriding focus needs to be on bringing in as much revenue as possible. The site-license or VPA avenue shouldn't be overlooked.

**Building the Installed Base.** Another benefit of site license programs is the building of a very important asset, your installed base of customers. A developer's installed base is an asset that, if nurtured effectively, can return untold dividends in the long run. An installed base of satisfied users creates a loyalty to the product that makes life difficult for the competition. The quicker you can build a large installed base (assuming that customers are satisfied with your product's performance), the stronger barrier against the competition you have constructed. This hopefully translates into a better chance for your company's long-term success.

Creating a large, loyal installed base leads to long-term relationships with customers, which has definite advantages. As your products permeate a customer's organization, they can become a critical part of that company's overall computing strategy. This gives you the ability to share ideas and information (prudently, of course)



about your long-term strategic direction and product focus with customers, so they can do better technology planning.

When this happens, the sales effort for your next product starts at a very early stage, because this relationship usually makes these customers want to be involved in the alpha and beta testing of your future products. When they see a need for these future products and are part of your feedback loop, they are more inclined to be early adopters of the products when they are commercially released. It also lets users give you input earlier in the development process, which can result in your being able to deliver products with maximum customer appeal.

This relationship can also provide you with excellent customer references. Quite often, while potential customers are evaluating your product, they ask questions about its large-scale performance—that is, how well it works when hundreds of connected people are using it, as is the case with Meeting Maker. If you can give prospects the names of a few very large satisfied customers, this may provide them with the impetus to close the deal.

IS (information systems) managers respect the opinions and advice of their counterparts at other companies much more than they trust the words of your sales reps. Having a loyal installed base to call upon as a reference is an asset that should be coveted.

Site licensing also helps reduce the potential of piracy. This is not to say that large companies knowingly make illegal copies, but a site license helps protect you against uninformed users who don't know they are doing wrong.

Looking at the big picture, I believe there are many benefits that you and your customers can get from a well-planned and executed site license/VPA program. It takes careful consideration to plan the process and excruciating attention to detail to get it right in the beginning, but if you do it for the right products in the right way, it can be well worth the effort.

John Shagoury is vice president of sales and marketing for On Technology, a software company based in Cambridge, Massachusetts that specializes in workgroup applications.

## **WHY CUSTOMERS ASK FOR SITE LICENSES**

To help decide what kind of site license agreement to create, you must first understand what arrangement will best meet the customer's needs. There are four major reasons why large customers typically favor a site license or VPA: better prices, user standardization, uncomplicated distribution, and reduced inventory.

**Price.** Price always seems to be an issue when you discuss large quantities. Many IS (information systems) or user-department managers have a fixed annual budget for software. Their job is to use that limited amount of money to generate the largest possible productivity increase through the cost-effective purchase of software and services. Cost-effectiveness: That's the motivation for a customer to want a site license.

An important factor when you're entering a site license price negotiation is whether the customer's software budget is managed centrally or departmentally. Centrally managed software budgets, usually administered by the IS department, typically

require that IS managers negotiate software agreements. In such a case, the IS department controls the budget and “user departments” have no budgetary responsibility for software purchases.

Centrally managed budgets usually present a much easier situation to deal with than the alternative—departmentally managed budgets. The latter usually means that site license negotiations occur with the corporate purchasing manager. This can frustrate developers, because the purchasing manager typically has little idea of the value (such as enhanced productivity) your software brings to the company.

Usually, the purchasing manager’s focus is only on price, terms, and conditions. This person has been assigned to make the purchase on the most favorable terms possible and often doesn’t truly differentiate between buying software and purchasing office supplies.

**Standardization.** Standardizing all users on a single application allows a company’s internal support organization to do a much better job. Any time a software product is purchased in volume for company-wide distribution, there will be a large percentage of novice and intermediate users, which potentially means a large internal support burden. During the last few years, corporate support organizations have remained relatively stagnant in terms of head count and resource growth, at the same time as the user base they support has grown tremendously. Four years ago, the average ratio of support person to users was perhaps 1:150, whereas today in many companies, it has increased to 1:400.

Standardizing on a specific application also allows IS organizations to enhance user productivity by more effectively developing and supporting customized templates and applications written around a standard piece of software. Not only can corporationwide processes (for creating and distributing forms, planning meetings, or doing expense reports, for example) be put into place, but users can then also share more information.

**Distribution.** Many large companies with centralized purchasing want to distribute software to users more effectively. There are various reasons for this, most of which make a site license attractive to them. One reason is that it gives the IS organization control over the implementation process.

For example, with our Meeting Maker product (a meeting scheduling tool that runs over networks) or an E-mail package, it is very typical for the IS organization to install the product in one department or one server group at a time. It can add 50 or 150 users per week and effectively manage the implementation process at a rate that won’t cause it, or the users, confusion or frustration. Alternatively, the IS procedure may mean installing the software individually on every user’s machine.

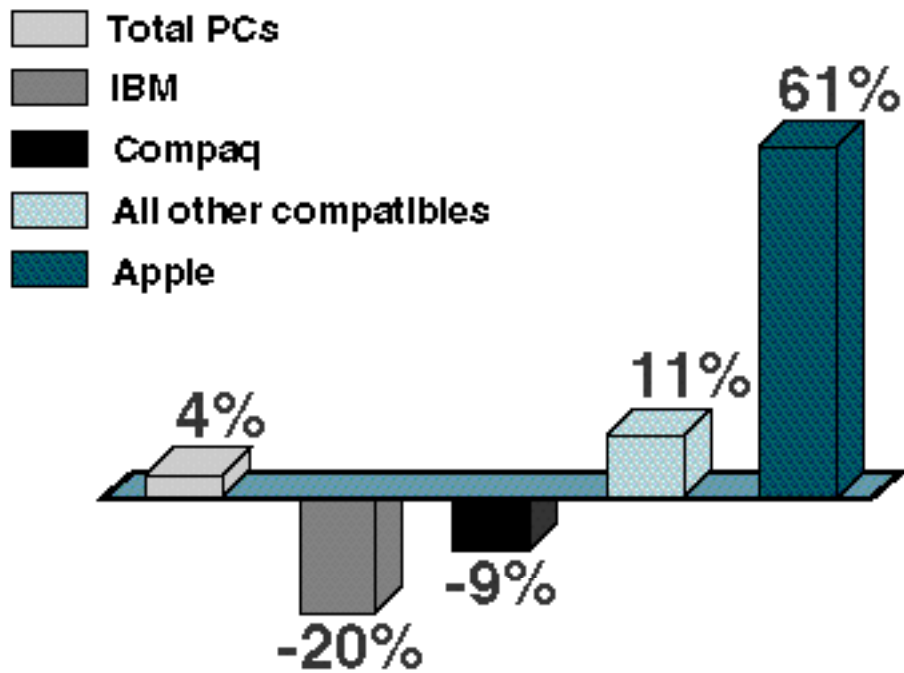
In either case, by controlling the distribution process, the IS people can make sure they get to every user, ensure that the software is installed correctly, and know that they have accurate records of which users have which software. Applying the adage “Knowledge is power,” the more the IS/ support organization knows about what software is being used, how, and by whom, the better it can support users and maximize their productivity.

**Inventory.** Another benefit corporate customers look for in site licensing is inventory reduction. Very few users need the packaging, and as we all know, very few use the manual. (One IS manager told us about a user who requested a manual for a site-licensed product. The IS manager knew that the user subsequently never looked at the documentation, because the IS manager gave the user a German manual—and the user never called back to request an English one.) Therefore, an IS manager wants to distribute software without needing a small warehouse to hold unopened or unneeded inventory.

This can be accomplished by giving the customer either a master of the application that can be downloaded from a server or a set of diskettes from which the IS manager lets staff members install the application on machines around the company.

In summary, if you can pin down exactly why your customers want a site license and the specific benefits it would render to their company, then you're much more likely to create the necessary win-win situation that keeps your revenue flowing and your customers happy.

### Percentage Change of PC Units Sold in 1990 vs. 1991



In the last several months in Apple Direct, we've been saying that Apple's recent unit growth has far outstripped the industry, an assertion that's borne out in the data above from Audits & Surveys, an independent research firm. The data shows the unit growth in the dealer channel from 1990 to 1991 in the months of January through September for both years. The figures speak for themselves.

Source: Audits & Surveys, Inc., 1991

## **Marketing Do's and Don'ts**

The Software Publishers Association has recently published a compilation of marketing tips entitled "The Do's and Don'ts of PC Software Marketing." The tips, which were gleaned from interviews with marketing professionals at 69 software companies, cover the gamut: from advertising, collateral, direct mail, and market research to guerilla marketing, pricing, and vertical marketing. In all, the guide contains more than 450 tips.

The SPA is offering Apple Direct readers a special price of \$99 for the guide (which is the same as the regular SPA member price and less than half the non-member price.) To order the guide, call Eileen Bramlet, SPA membership manager, at (202) 452-1600, ext. 326, or send her a fax at (202) 223-8756.

## **Apple direct**

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

Publication team:

Editor-in-Chief:

Lisa Raleigh (AppleLink:RALEIGH.L)

Technical Writer/Editor:

Gregg Williams (GREGGW)

Business and Departments Editor:

Dee Kiamy (KIAMY)

Production Manager & Designer:

Hartley G. Lesser (H.LESSER)

Contributors:

Jessa Vartanian, Suzanne Dills, Sharon Flowers, Monica Meffert, Todd Luchette, Kris Newby, Karen Wickre

Manager, DSSC:

David A. Krathwohl

Film:

Aptos Post, Aptos, CA

Prepress:

Prepress Assembly,  
San Francisco, CA

Printer:

Wolfer Press, Los Angeles, CA

APDA, Apple, the Apple logo, AppleLink, AppleShare, AppleTalk, A/UX, HyperCard, LaserWriter, Lisa, LocalTalk, Macintosh, MacApp, MacTCP, MPW, and StyleWriter are trademarks of Apple Computer, Inc., registered in the U.S. and other countries. AppleGlot, develop, Macintosh Quadra, MacX25, OneScanner, PhotoGrade, PowerBook, QuickTime, ResEdit, Sound Manager, SourceBug, SuperDrive, and TrueType are trademarks of Apple Computer, Inc. Classic is a registered trademark licensed to Apple Computer, Inc. PostScript is a trademark of Adobe Systems, Inc., registered in the U.S. and other countries. NuBus is a trademark of Texas Instruments.

UNIX is a registered trademark of UNIX Systems Laboratories. DECnet is a trademark of Digital Equipment Corp.

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