About This Book

This book, *Inside Macintosh: Interapplication Communication*, describes the interapplication communication architecture, which provides a standard and extensible mechanism for communication among Macintosh applications. This book also describes the system software routines that you can use to implement various forms of interapplication communication in your application.

If you are new to programming on the Macintosh computer, you should read *Inside Macintosh: Overview* for an introduction to general concepts of Macintosh programming; *Inside Macintosh: Macintosh Toolbox Essentials* for information on how to use menus, windows, and controls in your application; and *Macintosh Human Interface Guidelines* for a complete discussion of user interface guidelines and principles that every Macintosh application should follow.

This book describes how to implement publish and subscribe features in your application, how to communicate with other applications using Apple events, how to respond to scripts, and how to exchange information with other applications using the PPC Toolbox. It also discusses how your application can use the Data Access Manager to access information from a database application or other data source.

For an overview of all the features provided by the interapplication communication architecture, see the chapter "Introduction to Interapplication Communication" in this book.

To provide support for publish and subscribe features in your application, see the chapter "Edition Manager" in this book. This chapter describes how your application can allow users to share dynamic data among many documents.

To communicate with other applications by using Apple events, first see the chapter "Introduction to Apple Events" for a general introduction to Apple events. For information on how to respond to the required Apple events, see the chapter "Responding to Apple Events." To create and send Apple events, see the chapter "Creating and Sending Apple Events."

You can choose to write your application so that it can recognize descriptions, in Apple events, of objects in the application such as words, paragraphs, shapes, or documents. To do so, see the chapter "Resolving and Creating Object Specifier Records."

In addition to supporting Apple events, you can make your application scriptable—that is, capable of responding to Apple events sent to it by a scripting component. By executing scripts, users of scriptable applications can automate repetitive tasks or conditional tasks that involve multiple

applications. For more general information about scripting, see the chapter "Introduction to Scripting." See the chapter "Apple Event Terminology Resources" for information on the resources your application needs to provide in order to be scriptable.

You can also make your application recordable, that is, capable of recording a user's actions for later playback. For more information, see the chapter "Recording Apple Events."

For information on how your application can execute a script with the aid of a scripting component, see the chapter "Scripting Components."

Although you'll usually want to use Apple events to communicate with other applications, if you need low-level control of communication between applications you can use the Program-to-Program Communications (PPC) Toolbox. For more information, see the chapter "Program-to-Program Communications Toolbox."

Applications can use the Data Access Manager to access information from a database application or other data source. For example, a user in San Francisco might use a spreadsheet application to request data from a company database in New York. The spreadsheet application can use the Data Access Manager to request the data from the database. The database application in New York sends back the requested data, and the spreadsheet application can then use this data to generate a graph of the information. For information on sending and retrieving information from a data source, see the chapter "Data Access Manager."

For definitions of specific Apple events and Apple event objects, see the *Apple Event Registry: Standard Suites*, available from APDA.

For information on handling files in your application and a description of aliases and alias records, see *Inside Macintosh: Files*.

For information on processes and process serial numbers, see *Inside Macintosh: Processes*.

Format of a Typical Chapter

Almost all chapters in this book follow a standard structure. For example, the chapter "Creating and Sending Apple Events" contains these sections:

■ "Creating an Apple Event" and "Sending an Apple Event." These sections describe how your application can create and send Apple events. They describe the Apple Event Manager routines that you can use to accomplish these tasks, give related user interface information, and provide code samples and additional information.

■ "Reference to Creating and Sending Apple Events." This section provides a complete reference to the Apple Event Manager routines you can use to create and send Apple events. Each routine description also follows a

standard format, which presents the routine declaration followed by a description of every parameter of the routine. Some routine descriptions also give additional descriptive information, such as assembly-language information or result codes.

■ "Summary of Creating and Sending Apple Events." This section provides the Pascal and C interfaces for the constants, data structures, routines, and result codes associated with the Apple Event Manager routines for creating and sending Apple events. It also includes relevant assembly-language interface information.

Conventions Used in This Book

Inside Macintosh uses various conventions to present information. Words that require special treatment appear in specific fonts or font styles. Certain information, such as the contents of registers, use special formats so that you can scan them quickly.

Special Fonts

All code listings, reserved words, and the names of actual data structures, fields, constants, parameters, and routines are shown in Courier (`this is Courier`).

Words that appear in **boldface** are key terms or concepts and are defined in the Glossary.

Types of Notes

There are several types of notes used in this book.

Note
A note like this contains information that is interesting but possibly not essential to an understanding of the main text. (An example appears on page 3-26.) ◆

IMPORTANT
A note like this contains information that is essential for an understanding of the main text. (An example appears on page 3-33.) ▲

▲ **WARNING**
Warnings like this indicate potential problems that you should be aware of as you design your application. Failure to heed these warnings could result in system crashes or loss of data. (An example appears on page page 4-10.) ▲

Assembly-Language Information

Some chapters provide additional assembly-language information. For example, *Inside Macintosh* provides information about the registers for specific routines like this:

Registers on entry

A0   Contents of register A0 on entry

Registers on exit

D0   Contents of register D0 on exit

In addition, *Inside Macintosh* provides information about the fields of a parameter block in this format:

↔	`inAndOut`	`Integer`	Input/output parameter.
←	`output1`	`Ptr`	Output parameter.
→	`input1`	`Ptr`	Input parameter.

The arrow in the far left column indicates whether the field is an input parameter, output parameter, or both. You must supply values for all input parameters and input/output parameters. The routine returns values in output parameters and input/output parameters.

The second column shows the field name as defined in the MPW Pascal interface files; the third column indicates the Pascal data type of that field. The fourth column provides a brief description of the use of the field. For a complete description of each field, see the discussion that follows the parameter block or the description of the parameter block in the reference section of the chapter.

The Development Environment

The system software routines described in this book are available using Pascal, C, or assembly-language interfaces. How you access these routines depends on the development environment you are using. When showing system software routines, this book uses the Pascal interface available with the Macintosh Programmer's Workshop (MPW).

All code listings in this book are shown in Pascal (except for listings that describe resources, which are shown in Rez-input format). They show methods of using various routines and illustrate techniques for accomplishing particular tasks. All code listings have been compiled and, in many cases, tested. However, Apple Computer, Inc., does not intend for you to use these code samples in your application. You can find the location of code listings in the list of figures, tables, and listings. If you know the name of a particular

routine (such as `MyHandleODoc` or `MyHandleQuit`) shown in a code listing, you can find the page on which the routine occurs by looking under the entry "sample routines" in the index of this book.

In order to make the code listings in this book more readable, they show only limited error handling. You need to develop your own techniques for handling errors.

This book occasionally illustrates concepts by reference to sample applications called *SurfWriter, SurfDB,* and *SurfCharter;* these are not actual products of Apple Computer, Inc.

APDA is Apple's worldwide source for over three hundred development tools, technical resources, training products, and information for anyone interested in developing applications on Apple platforms. Customers receive the quarterly *APDA Tools Catalog* featuring all current versions of Apple development tools and the most popular third-party development tools. Ordering is easy; there are no membership fees, and application forms are not required for most of our products. APDA offers convenient payment and shipping options, including site licensing.

To order products or to request a complimentary copy of the *APDA Tools Catalog*, contact

APDA
Apple Computer, Inc.
P.O. Box 319
Buffalo, NY 14207-0319

Telephone	800-282-2732 (United States)
	800-637-0029 (Canada)
	716-871-6555 (International)
Fax	716-871-6511
AppleLink	APDA
America Online	APDA
CompuServe	76666,2405
Internet	APDA@applelink.apple.com

If you provide commercial products and services, call 408-974-4897 for information on the developer support programs available from Apple.

For information on registering signatures, file types, and other technical information, contact

Macintosh Developer Technical Support
Apple Computer, Inc.
20525 Mariani Avenue, M/S 75-3T
Cupertino, CA 95014-6299

The Apple Event Registrar maintains the *Apple Event Registry: Standard Suites* and other information about the ongoing development of Apple event suites. For more information about Apple event suites, including those under development, send electronic mail to the AppleLink address REGISTRY.