PREFACE

About This Book

This book, *Inside Macintosh: Sound,* describes the parts of the Macintosh system software that allow you to manage sounds. It describes the services provided by the three principal sound-related system software managers (the Sound Manager, the Sound Input Manager, and the Speech Manager) and shows in detail how your application can record and play back sounds, compress and expand audio data, convert text to speech, and perform other similar operations.

If you are not yet experienced with playing or recording sounds on Macintosh computers, you should begin with the chapter "Introduction to Sound on the Macintosh." That chapter describes the services provided by the system software and shows how to use the most basic sound-related capabilities of Macintosh computers. It provides complete source code examples illustrating how to record sounds into resources and files, how to play sounds stored in resources and files, and how to convert written text into spoken words. It's possible that this introductory chapter contains all the information you need to successfully integrate sound into your application.

Once you are familiar with basic sound recording and production on Macintosh computers, you might want to read other chapters in this book. The chapter "Sound Manager" provides complete information about sound output. It shows how to control sound production at a very low level, how to produce sound asynchronously (that is, while other operations in the computer take place), and how to compress and expand audio data. This chapter also provides complete details about the structure of the two main sound storage formats, sound resources and sound files.

If you need more control over the sound recording process than is offered by the basic recording functions described in the chapter "Introduction to Sound on the Macintosh," you need to read the chapter "Sound Input Manager." That chapter shows how to record sound without displaying the sound recording dialog box or to interact directly with a sound input device driver.

The chapter "Speech Manager" shows how you can convert written text into speech. You'll need to read this chapter if you want to convert arbitrary blocks of text (such as very large buffers of text) into spoken words, or if you need to gain very fine control over speech production (for example, to synchronize speech production with other activities, or to use customized pronunciation dictionaries).

The chapter "Sound Components" describes how to write sound components. The Sound Manager uses sound components to manipulate audio data or to communicate with sound output devices. You need to read this chapter only if you are developing a new sound output device or want to use a custom audio data compression and expansion scheme.

The chapter “Audio Components” describes how to write audio components. The Sound Manager uses audio components to adjust volumes or other settings of a sound output device when the device contains multiple output ports that can be independently controlled by software. You need to read this chapter only if you are developing a new sound output device that contains several sound-producing ports (such as both speakers and headphones).

Format of a Typical Chapter

Almost all chapters in this book follow a standard structure. For example, the chapter “Sound Input Manager” contains these sections:

- “About the Sound Input Manager.” This section provides an overview of the features provided by the Sound Input Manager.
- “Using the Sound Input Manager.” This section describes the tasks you can accomplish using the Sound Input Manager. It describes how to use the most common routines, gives related user interface information, provides code samples, and supplies additional information.
- “Sound Input Manager Reference.” This section provides a complete reference for the Sound Input Manager by describing the constants, data structures, routines, and resources it uses. 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 the Sound Input Manager.” This section provides the Pascal and C interfaces for the constants, data structures, routines, and result codes associated with the Sound Input Manager. It also includes relevant assembly-language interface information.

Conventions Used in This Book

Inside Macintosh uses special conventions to present certain types of information. Words that require special treatment appear in specific fonts or font styles. Certain information, such as parameter blocks, appears in special formats so that you can scan it quickly.

Special Fonts

All code listings, reserved words, and the names of actual data structures, constants, fields, 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 1-6.) ◆

IMPORTANT
A note like this contains information that is essential for an understanding of the main text. (An example appears on page 1-9.) ▲

▲ **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 2-24.) ▲

Assembly-Language Information

Inside Macintosh provides information about the registers for specific routines in this format:

Registers on entry

A0	Contents of register A0 on entry

Registers on exit

D0	Contents of register D0 on exit

In the "Assembly-Language Summary" section at the end of each chapter, *Inside Macintosh* presents information about the fields of data structures in this format:

0	`what`	word	event code
2	`message`	long	event message
6	`when`	long	ticks since startup

The left column indicates the byte offset of the field from the beginning of the data structure. The second column shows the field name as defined in the MPW Pascal interface files; the third column indicates the size 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 of the data structure in the reference section of the chapter.

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

Parameter block

↔	`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.

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 interfaces available with the Macintosh Programmer’s Workshop (MPW). However, the chapters “Sound Components” and “Audio Components” use C interfaces, because Pascal interfaces are not currently available.

All code listings in this book are shown in Pascal or C. They show methods of using various routines and illustrate techniques for accomplishing particular tasks. All code listings have been compiled and, in most cases, tested. However, Apple Computer, Inc. does not intend for you to use these code samples in your application.

This book occasionally illustrates concepts by referring to a sample application called SurfWriter. This application is not an actual product of Apple Computer, Inc. This book also uses the names SurfBoard and WaveMaker to refer to sample sound output and input devices. These devices are not actual products of Apple Computer, Inc.

For More Information

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 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	APDAorder
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 of registering signatures, file types, and other technical information, contact

Macintosh Developer Technical Support
Apple Computer, Inc.
20525 Mariani Avenue, M/S 303-2T
Cupertino, CA 95014-6299