About This Book

This book, *Inside Macintosh: Memory,* describes the parts of the Macintosh Operating System that allow you to allocate memory directly, release it, or otherwise manipulate it. The book includes introductory material about managing memory on Macintosh computers as well as a complete technical reference to the Memory Manager, the Virtual Memory Manager, and other memory-related services provided by the system software.

If you are new to programming on the Macintosh Operating System, you should begin with the chapter "Introduction to Memory Management." This chapter provides a general introduction to memory management on Macintosh computers. It describes how the Operating System organizes and manages the available memory, and it shows how you can use the services provided by the Memory Manager and other system software components to manage the memory in your application partition effectively. Because this chapter is designed to be largely self-contained, the reference and summary sections in this chapter are subsets of the corresponding sections from the other chapters in this book.

Once you are familiar with basic memory management on Macintosh computers, you should look at the chapter "Memory Manager." It describes how to allocate memory outside your application partition and how to perform more advanced memory operations than are described in the introductory chapter.

The chapter "Virtual Memory Manager" describes the operation of virtual memory and describes the routines that you can use to intervene in the otherwise automatic operations of the Virtual Memory Manager. Most applications are not affected by the operation of virtual memory and do not need to use the routines provided by the Virtual Memory Manager. If your application sends memory addresses to some NuBus™ master hardware, however, you should read the discussion of mapping virtual to physical addresses in that chapter.

The final chapter in this book, "Memory Management Utilities," describes a number of utility routines provided by the system software. You need to read this chapter primarily if you install routines that are executed by system software routines or in response to an interrupt, or if you need to change the addressing mode. You also need to read this chapter if your application might be affected by the normal operation of the processor's instruction or data caches.

Format of a Typical Chapter

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

- "About the Memory Manager." This section provides an overview of the features provided by the Memory Manager.
- "Using the Memory Manager." This section describes the tasks you can accomplish using the Memory Manager. It describes how to use the most common routines, gives related user interface information, provides code samples, and supplies additional information.
- "Memory Manager Reference." This section provides a complete reference to the Memory Manager by describing the data structures, routines, and resources that it uses. Each routine description also follows a standard format, which gives the routine declaration and 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 Memory Manager." This section provides the Memory Manager's Pascal interface, as well as the C interface, for the constants, data structures, routines, and result codes associated with the Memory Manager. It also includes relevant assembly-language interface information.

Some chapters also contain additional main sections that provide more detailed discussions of certain topics. For example, the Memory Manager chapter contains the section "Organization of Memory" that describes how the Memory Manager organizes zones and blocks in RAM.

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 parameter blocks, use special formats so that you can scan them 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 1-8.) ◆

IMPORTANT

A note like this contains information that is essential for an understanding of the main text. (An example appears on page 2-7.) ▲

▲ **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 1-16.) ▲

Assembly-Language Information

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* 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. This book shows system software routines in their Pascal interface using the Macintosh Programmer's Workshop (MPW).

All code listings in this book are shown in Pascal. 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 does not intend that you use these code samples in your application.

APDA, Apple's source for developer tools, offers worldwide access to a broad range of programming products, resources, and information for anyone developing on Apple platforms. You'll find the most current versions of Apple and third-party development tools, debuggers, compilers, languages, and technical references for all Apple platforms. To establish an APDA account, obtain additional ordering information, or find out about site licensing and developer training programs, contact

APDA
Apple Computer, Inc.
20525 Mariani Avenue, M/S 33-G
Cupertino, CA 95014-6299

Telephone:	800-282-2732 (United States) 800-637-0029 (Canada) 800-562-3910 (elsewhere in the world)
Fax:	408-562-3971
Telex:	171-576

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, Apple events, and other technical information, contact

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