# About This Book

This book describes the parts of the Macintosh Operating System that allow you to manage processes and tasks. It includes introductory material about managing processes on Macintosh computers as well as a complete technical reference to the Process Manager, the Time Manager, the Vertical Retrace Manager, and other process-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 Processes and Tasks." This chapter provides a general introduction to process and task management on Macintosh computers. It describes how the Operating System controls access to the CPU and other system resources to create a cooperative multitasking environment in which your application and any other open applications run. This environment is managed primarily by the Process Manager, which is responsible for launching processes, scheduling their use of the available system resources, and handling their termination.

This chapter also describes how your application can install tasks that are executed asynchronously from your application, usually in response to interrupts. You can

- execute a task after a certain amount of time has elapsed
- execute a task repetitively
- notify the user while your application is in the background
- execute a task between screen refreshes
- execute a routine as part of the shutdown or restart process

Once you are familiar with basic process and task management on Macintosh computers, you might need to look at the chapter "Process Manager." It describes how you can get information about open processes and, if necessary, launch processes and desk accessories. You can also use the Process Manager to alter the processing status of an application or to terminate your application.

If you want a task to be executed after some specified amount of time has elapsed, you can use the Time Manager to schedule that task for later execution. The task can reschedule itself, so you can use the Time Manager to execute a routine repetitively. You can also use the Time Manager to calculate elapsed times and to synchronize events in the Macintosh computer. See the chapter "Time Manager" for details.

The Vertical Retrace Manager allows you to schedule a task for execution during vertical retrace interrupts. Like Time Manager tasks, vertical retrace tasks can reschedule themselves so that they are executed repetitively. In general, you should use the Vertical Retrace Manager to handle repetitive tasks that need to be synchronized with the redrawing of the screen and the Time Manager to handle those tasks that don't. See the chapter "Vertical Retrace Manager" for details.

You can use the Notification Manager to inform users of significant occurrences in applications that are running in the background or in software that is largely invisible to the user. This software includes device drivers, vertical blanking (VBL) tasks, Time Manager tasks, completion routines, and desk accessories that operate behind the scenes. See the chapter "Notification Manager" for complete details.

You should read the chapter "Segment Manager" for information about how the Operating System manages the loading and unloading of your application's code segments into and out of memory. By dividing your application's executable code into segments, you allow it to run in a memory partition that is smaller than the total size of the application itself.

The final chapter in this book, "Shutdown Manager," shows how you can install procedures that are executed as part of the final stages of shutting down or restarting a Macintosh computer.

## Format of a Typical Chapter

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

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

Some chapters contain additional main sections that provide more detailed discussions of certain topics. For example, in the chapter "Shutdown Manager," the section "The Shutdown Process" describes the process that the Shutdown Manager procedures perform to shut down or restart the system.

# 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 1-4.) ◆

**IMPORTANT**
A note like this contains information that is essential for an understanding of the main text. (An example appears on page 2-16.) ▲

▲ **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 1-12.) ▲

## 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.

This book occasionally uses *SurfWriter* as the name of a sample application for illustrative purposes; this is not an actual product of Apple Computer, Inc.

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)<br>800-637-0029 (Canada)<br>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