

INSIDE MACINTOSH

Memory



Addison-Wesley Publishing Company

Reading, Massachusetts Menlo Park, California New York
Don Mills, Ontario Wokingham, England Amsterdam Bonn
Sydney Singapore Tokyo Madrid San Juan
Paris Seoul Milan Mexico City Taipei

Apple Computer, Inc.
© 1992, Apple Computer, Inc.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Computer, Inc. Printed in the United States of America.

No licenses, express or implied, are granted with respect to any of the technology described in this book. Apple retains all intellectual property rights associated with the technology described in this book. This book is intended to assist application developers to develop applications only for Apple Macintosh computers.

Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, CA 95014-6299
408-996-1010

Apple, the Apple logo, APDA, AppleShare, A/UX, LaserWriter, Macintosh, MPW, and MultiFinder are trademarks of Apple Computer, Inc., registered in the United States and other countries.

Finder, PowerBook, and QuickDraw are trademarks of Apple Computer, Inc.

Adobe Illustrator and PostScript are trademarks of Adobe Systems Incorporated, which may be registered in certain jurisdictions.

AGFA is a trademark of Agfa-Gevaert.

FrameMaker is a registered trademark of Frame Technology Corporation.

Helvetica and Palatino are registered trademarks of Linotype Company.

ITC Zapf Dingbats is a registered trademark of International Typeface Corporation.

NuBus is a trademark of Texas Instruments.

Simultaneously published in the United States and Canada.

LIMITED WARRANTY ON MEDIA AND REPLACEMENT

ALL IMPLIED WARRANTIES ON THIS MANUAL, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO NINETY (90) DAYS FROM THE DATE OF THE ORIGINAL RETAIL PURCHASE OF THIS PRODUCT.

Even though Apple has reviewed this manual, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS MANUAL IS SOLD "AS IS," AND YOU, THE PURCHASER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS MANUAL, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

ISBN 0-201-63240-3
1 2 3 4 5 6 7 8 9-MU-9695949392
First Printing, August 1992

Contents

	Figures, Tables, and Listings	ix
Preface	About This Book	xi
<hr/>		
	Format of a Typical Chapter	xii
	Conventions Used in This Book	xii
	Special Fonts	xii
	Types of Notes	xiii
	Assembly-Language Information	xiii
	Development Environment	xiv
<hr/>		
Chapter 1	Introduction to Memory Management	1-1
<hr/>		
	About Memory	1-4
	Organization of Memory by the Operating System	1-4
	The System Heap	1-6
	The System Global Variables	1-6
	Organization of Memory in an Application Partition	1-7
	The Application Stack	1-8
	The Application Heap	1-9
	The Application Global Variables and A5 World	1-12
	Temporary Memory	1-13
	Virtual Memory	1-15
	Addressing Modes	1-15
	Heap Management	1-16
	Relocatable and Nonrelocatable Blocks	1-16
	Properties of Relocatable Blocks	1-20
	Locking and Unlocking Relocatable Blocks	1-20
	Purging and Reallocating Relocatable Blocks	1-21
	Memory Reservation	1-22
	Heap Purging and Compaction	1-23
	Heap Fragmentation	1-24
	Deallocating Nonrelocatable Blocks	1-25
	Reserving Memory	1-25
	Locking Relocatable Blocks	1-26
	Allocating Nonrelocatable Blocks	1-27
	Summary of Preventing Fragmentation	1-28
	Dangling Pointers	1-29
	Compiler Dereferencing	1-29
	Loading Code Segments	1-31
	Callback Routines	1-32

Invalid Handles	1-33	
Disposed Handles	1-33	
Empty Handles	1-34	
Fake Handles	1-35	
Low-Memory Conditions	1-36	
Memory Cushions	1-36	
Memory Reserves	1-37	
Grow-Zone Functions	1-38	
Using Memory	1-38	
Setting Up the Application Heap	1-38	
Changing the Size of the Stack	1-39	
Expanding the Heap	1-40	
Allocating Master Pointer Blocks	1-41	
Determining the Amount of Free Memory	1-42	
Allocating Blocks of Memory	1-44	
Maintaining a Memory Reserve	1-46	
Defining a Grow-Zone Function	1-48	
Memory Management Reference	1-50	
Memory Management Routines	1-50	
Setting Up the Application Heap	1-50	
Allocating and Releasing Relocatable Blocks of Memory	1-54	
Allocating and Releasing Nonrelocatable Blocks of Memory	1-58	
Setting the Properties of Relocatable Blocks	1-60	
Managing Relocatable Blocks	1-67	
Manipulating Blocks of Memory	1-73	
Assessing Memory Conditions	1-75	
Grow-Zone Operations	1-77	
Setting and Restoring the A5 Register	1-78	
Application-Defined Routines	1-80	
Grow-Zone Functions	1-80	
Summary of Memory Management	1-82	
Pascal Summary	1-82	
Data Types	1-82	
Memory Management Routines	1-82	
Application-Defined Routines	1-83	
C Summary	1-84	
Data Types	1-84	
Memory Management Routines	1-84	
Application-Defined Routines	1-85	
Assembly-Language Summary	1-86	
Global Variables	1-86	
Result Codes	1-86	

About the Memory Manager	2-3
Temporary Memory	2-4
Multiple Heap Zones	2-5
The System Global Variables	2-6
Using the Memory Manager	2-7
Reading and Writing System Global Variables	2-8
Extending an Application's Memory	2-9
Allocating Temporary Memory	2-10
Determining the Features of Temporary Memory	2-11
Using the System Heap	2-12
Allocating Memory at Startup Time	2-13
Creating Heap Zones	2-14
Installing a Purge-Warning Procedure	2-16
Organization of Memory	2-19
Heap Zones	2-19
Block Headers	2-22
Memory Manager Reference	2-24
Data Types	2-24
Memory Manager Routines	2-26
Setting Up the Application Heap	2-27
Allocating and Releasing Relocatable Blocks of Memory	2-29
Allocating and Releasing Nonrelocatable Blocks of Memory	2-35
Changing the Sizes of Relocatable and Nonrelocatable Blocks	2-39
Setting the Properties of Relocatable Blocks	2-43
Managing Relocatable Blocks	2-51
Manipulating Blocks of Memory	2-59
Assessing Memory Conditions	2-66
Freeing Memory	2-71
Grow-Zone Operations	2-76
Allocating Temporary Memory	2-77
Accessing Heap Zones	2-80
Manipulating Heap Zones	2-83
Application-Defined Routines	2-89
Grow-Zone Functions	2-89
Purge-Warning Procedures	2-90
Summary of the Memory Manager	2-93
Pascal Summary	2-93
Constants	2-93
Data Types	2-93
Memory Manager Routines	2-94
Application-Defined Routines	2-97
C Summary	2-97
Constants	2-97
Data Types	2-97
Memory Manager Routines	2-98

Application-Defined Routines	2-101
Assembly-Language Summary	2-101
Constants	2-101
Data Structures	2-102
Trap Macros	2-102
Global Variables	2-104
Result Codes	2-105

Chapter 3

Virtual Memory Manager 3-1

About the Virtual Memory Manager	3-3
Virtual Memory	3-4
The Logical Address Space	3-5
24-Bit Addressing	3-5
32-Bit Addressing	3-7
The Physical Address Space	3-9
Page Faults	3-11
Using the Virtual Memory Manager	3-13
Obtaining Information About Virtual Memory	3-14
Holding and Releasing Memory	3-14
Locking and Unlocking Memory	3-15
Mapping Logical to Physical Addresses	3-16
Deferring User Interrupt Handling	3-20
Virtual Memory and Debuggers	3-21
Bus-Error Vectors	3-22
Special Nonmaskable Interrupt Needs	3-22
Supervisor Mode	3-23
The Debugging State	3-23
Keyboard Input	3-23
Page States	3-24
Virtual Memory Manager Reference	3-24
Data Structures	3-24
Memory-Block Record	3-24
Translation Table	3-25
Routines	3-25
Virtual Memory Management	3-25
Virtual Memory Debugger Support Routines	3-34
Summary of the Virtual Memory Manager	3-41
Pascal Summary	3-41
Constants	3-41
Data Types	3-41
Routines	3-42

C Summary	3-42
Constants	3-42
Data Types	3-43
Routines	3-43
Assembly-Language Summary	3-44
Data Types	3-44
Trap Macros	3-44
Result Codes	3-45

Chapter 4	Memory Management Utilities	4-1
-----------	------------------------------------	-----

The Memory Control Panel	4-3
About the Memory Management Utilities	4-5
The A5 Register	4-5
Addressing Modes	4-7
Address Translation	4-8
Processor Caches	4-8
Stale Instructions	4-9
Stale Data	4-10
Using the Memory Management Utilities	4-13
Accessing the A5 World in Completion Routines	4-14
Accessing the A5 World in Interrupt Tasks	4-16
Using QuickDraw Global Variables in Stand-Alone Code	4-18
Switching Addressing Modes	4-20
Stripping Flag Bits From Memory Addresses	4-21
Translating Memory Addresses	4-23
Memory Management Utilities Reference	4-24
Routines	4-24
Setting and Restoring the A5 Register	4-24
Changing the Addressing Mode	4-26
Manipulating Memory Addresses	4-27
Manipulating the Processor Caches	4-29
Summary of the Memory Management Utilities	4-34
Pascal Summary	4-34
Constants	4-34
Routines	4-34
C Summary	4-35
Constants	4-35
Routines	4-35
Assembly-Language Summary	4-36
Trap Macros	4-36
Global Variables	4-36
Result Codes	4-36

Glossary GL-1

Index IN-1

Figures, Tables, and Listings

Chapter 1

Introduction to Memory Management 1-1

Figure 1-1	Memory organization with several applications open	1-5
Figure 1-2	Organization of an application partition	1-7
Figure 1-3	The application stack	1-9
Figure 1-4	A fragmented heap	1-10
Figure 1-5	A compacted heap	1-11
Figure 1-6	Organization of an application's A5 world	1-12
Figure 1-7	Using temporary memory allocated from unused RAM	1-14
Figure 1-8	A pointer to a nonrelocatable block	1-17
Figure 1-9	A handle to a relocatable block	1-19
Figure 1-10	Purging and reallocating a relocatable block	1-22
Figure 1-11	Allocating a nonrelocatable block	1-23
Figure 1-12	An effectively partitioned heap	1-26
Listing 1-1	Locking a block to avoid dangling pointers	1-30
Listing 1-2	Creating a fake handle	1-35
Listing 1-3	Increasing the amount of space allocated for the stack	1-40
Listing 1-4	Setting up your application heap and stack	1-42
Listing 1-5	Determining whether allocating memory would deplete the memory cushion	1-43
Listing 1-6	Allocating relocatable blocks	1-44
Listing 1-7	Allocating nonrelocatable blocks	1-45
Listing 1-8	Allocating a dialog record	1-45
Listing 1-9	Creating an emergency memory reserve	1-46
Listing 1-10	Checking the emergency memory reserve	1-47
Listing 1-11	Determining whether allocating memory would deplete the memory cushion	1-47
Listing 1-12	Reallocating the emergency memory reserve	1-48
Listing 1-13	A grow-zone function that releases emergency storage	1-49

Chapter 2

Memory Manager 2-1

Figure 2-1	A block header in a 24-bit zone	2-22
Figure 2-2	A block header in a 32-bit zone	2-23
Listing 2-1	Reading the value of a system global variable	2-8
Listing 2-2	Changing the value of a system global variable	2-9
Listing 2-3	Determining whether temporary-memory routines are available	2-12
Listing 2-4	Calling a procedure by address	2-13
Listing 2-5	Creating a subzone of the original application heap zone	2-15
Listing 2-6	A purge-warning procedure	2-17
Listing 2-7	Installing a purge-warning procedure	2-18
Listing 2-8	A purge-warning procedure that calls the Resource Manager's procedure	2-19

Chapter 3

Virtual Memory Manager 3-1

- Figure 3-1** 24-bit Memory Manager logical address space 3-6
- Figure 3-2** 32-bit Memory Manager logical address space 3-8
- Figure 3-3** The physical address space on a Macintosh IIci with 8 MB of RAM 3-10

- Listing 3-1** Translating logical to physical addresses 3-19

Chapter 4

Memory Management Utilities 4-1

- Figure 4-1** The Memory control panel 4-4
- Figure 4-2** Initializing a status code 4-11
- Figure 4-3** Reading stale data 4-12
- Figure 4-4** Reading invalid instructions 4-13

- Table 4-1** Caches available in MC680x0 microprocessors 4-9

- Listing 4-1** A sample grow-zone function 4-15
- Listing 4-2** Passing A5 to a notification response procedure 4-16
- Listing 4-3** Setting up and restoring the A5 register at interrupt time 4-17
- Listing 4-4** Structure of the QuickDraw global variables 4-18
- Listing 4-5** Copying the QuickDraw global variables into a record 4-19
- Listing 4-6** A control's draw routine using the calling application's QuickDraw patterns 4-19

- Listing 4-7** Stripping the program counter 4-21
- Listing 4-8** Stripping addresses in time-critical code 4-23
- Listing 4-9** Calculating the `StripAddress` mask 4-23
- Listing 4-10** Translating 24-bit to 32-bit addresses 4-24

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.

PREFACE

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

PREFACE

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