

# INSIDE MACINTOSH

---

## Imaging With QuickDraw



**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.  
© 1994 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.

Every effort has been made to ensure that the information in this manual is accurate. Apple is not responsible for printing or clerical errors.

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

Apple, the Apple logo, APDA, AppleLink, AppleTalk, ImageWriter, LaserWriter, Macintosh, MPW, PowerBook, and StyleWriter are trademarks of Apple Computer, Inc., registered in the United States and other countries.

ColorSync, Finder, Geneva, QuickDraw, QuickTime, ResEdit, System 7, and TrueType are trademarks of Apple Computer, Inc.

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

America Online is a service mark of Quantum Computer Services, Inc.

Classic is a registered trademark licensed to Apple Computer, Inc.

CompuServe is a registered service mark of CompuServe, Inc.

FrameMaker is a registered trademark of Frame Technology Corporation.

Helvetica and Palatino are registered trademarks of Linotype Company.

Internet is a trademark of Digital Equipment Corporation.

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

MacPaint is a registered trademark of Claris Corporation.

NuBus is a trademark of Texas Instruments.

Motorola is a registered trademark of Motorola Corporation.

Optrotech is a trademark of Orbotech Corporation.

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 0-201-63242-X  
1 2 3 4 5 6 7 8 9-CRW-9897969594  
First Printing, March 1994

The paper used in this book meets the EPA standards for recycled fiber.

#### Library of Congress Cataloging-in-Publication Data

Inside Macintosh. Imaging with QuickDraw / Apple Computer, Inc.

p. cm.

Includes index.

ISBN 0-201-63242-X

1. Macintosh (Computer)—Programming. 2. Computer graphics.  
3. QuickDraw. I. Title: Imaging with QuickDraw.

QA76.8.M3I46 1994

006.6'765—dc20

93-50183  
CIP

# Contents

	Figures, Tables, and Listings	xiii
Preface	About This Book	xxi
<hr/>		
	Format of a Typical Chapter	xxii
	Conventions Used in This Book	xxiii
	Special Fonts	xxiii
	Types of Notes	xxiii
	Empty Strings	xxiv
	Assembly-Language Information	xxiv
	The Development Environment	xxiv
Chapter 1	Introduction to QuickDraw	1-1
<hr/>		
	Drawing Environments	1-4
	QuickDraw's Coordinate Plane	1-6
	Images	1-10
	Colors	1-17
	Indexed Colors	1-19
	Direct Colors	1-20
	Multiple Screens	1-21
	From Memory Bits to Onscreen Pixels	1-24
	From Memory Bits to Printers	1-26
	Other Graphics Managers	1-28
Chapter 2	Basic QuickDraw	2-1
<hr/>		
	About Basic QuickDraw	2-3
	The Mathematical Foundations of QuickDraw	2-4
	The Coordinate Plane	2-4
	Points	2-4
	Rectangles	2-5
	Regions	2-7
	The Black-and-White Drawing Environment: Basic Graphics Ports	2-7
	Bitmaps	2-9
	The Graphics Port Drawing Area	2-11
	Graphics Port Bit Patterns	2-13
	The Graphics Pen	2-13
	Text in a Graphics Port	2-13

The Limited Colors of a Basic Graphics Port	2-14
Other Fields	2-14
Using Basic QuickDraw	2-14
Initializing Basic QuickDraw	2-16
Creating Basic Graphics Ports	2-16
Setting the Graphics Port	2-18
Switching Between Global and Local Coordinate Systems	2-19
Scrolling the Pixels in the Port Rectangle	2-20
Basic QuickDraw Reference	2-26
Data Structures	2-26
Routines	2-36
Initializing QuickDraw	2-36
Opening and Closing Basic Graphics Ports	2-37
Saving and Restoring Graphics Ports	2-41
Managing Bitmaps, Port Rectangles, and Clipping Regions	2-43
Manipulating Points in Graphics Ports	2-51
Summary of Basic QuickDraw	2-56
Pascal Summary	2-56
Data Types	2-56
Routines	2-57
C Summary	2-58
Data Types	2-58
Functions	2-60
Assembly-Language Summary	2-61
Data Structures	2-61
Global Variables	2-62
Result Codes	2-62

## Chapter 3

## QuickDraw Drawing 3-1

---

About QuickDraw Drawing	3-3
The Graphics Pen	3-4
Bit Patterns	3-5
Boolean Transfer Modes With 1-Bit Pixels	3-8
Lines and Shapes	3-11
Defining Lines and Shapes	3-11
Framing Shapes	3-12
Painting and Filling Shapes	3-12
Erasing Shapes	3-12
Inverting Shapes	3-13
Other Graphic Entities	3-13
The Eight Basic QuickDraw Colors	3-14
Drawing With QuickDraw	3-16
Drawing Lines	3-17
Drawing Rectangles	3-22

Drawing Ovals, Arcs, and Wedges	3-25
Drawing Regions and Polygons	3-27
Performing Calculations and Other Manipulations of Shapes	3-31
Copying Bits Between Graphics Ports	3-32
Customizing QuickDraw's Low-Level Routines	3-35
QuickDraw Drawing Reference	3-36
Data Structures	3-36
Routines	3-41
Managing the Graphics Pen	3-41
Changing the Background Bit Pattern	3-48
Drawing Lines	3-49
Creating and Managing Rectangles	3-52
Drawing Rectangles	3-58
Drawing Rounded Rectangles	3-63
Drawing Ovals	3-68
Drawing Arcs and Wedges	3-71
Creating and Managing Polygons	3-78
Drawing Polygons	3-81
Creating and Managing Regions	3-85
Drawing Regions	3-100
Scaling and Mapping Points, Rectangles, Polygons, and Regions	3-104
Calculating Black-and-White Fills	3-108
Copying Images	3-112
Drawing With the Eight-Color System	3-122
Determining Whether QuickDraw Has Finished Drawing	3-125
Getting Pattern Resources	3-126
Customizing QuickDraw Operations	3-129
Resources	3-140
The Pattern Resource	3-140
The Pattern List Resource	3-141
Summary of QuickDraw Drawing	3-142
Pascal Summary	3-142
Constants	3-142
Data Types	3-144
Routines	3-145
C Summary	3-149
Constants	3-149
Data Types	3-151
Functions	3-152
Assembly-Language Summary	3-157
Data Structures	3-157
Global Variables	3-158

---

About Color QuickDraw	4-4
RGB Colors	4-4
The Color Drawing Environment: Color Graphics Ports	4-5
Pixel Maps	4-9
Pixel Patterns	4-12
Color QuickDraw's Translation of RGB Colors to Pixel Values	4-13
Colors on Grayscale Screens	4-17
Using Color QuickDraw	4-18
Initializing Color QuickDraw	4-19
Creating Color Graphics Ports	4-20
Drawing With Different Foreground Colors	4-21
Drawing With Pixel Patterns	4-23
Copying Pixels Between Color Graphics Ports	4-26
Boolean Transfer Modes With Color Pixels	4-32
Dithering	4-37
Arithmetic Transfer Modes	4-38
Highlighting	4-41
Color QuickDraw Reference	4-44
Data Structures	4-45
Color QuickDraw Routines	4-63
Opening and Closing Color Graphics Ports	4-63
Managing a Color Graphics Pen	4-67
Changing the Background Pixel Pattern	4-68
Drawing With Color QuickDraw Colors	4-70
Determining Current Colors and Best Intermediate Colors	4-79
Calculating Color Fills	4-82
Creating, Setting, and Disposing of Pixel Maps	4-85
Creating and Disposing of Pixel Patterns	4-87
Creating and Disposing of Color Tables	4-91
Retrieving Color QuickDraw Result Codes	4-94
Customizing Color QuickDraw Operations	4-96
Reporting Data Structure Changes to QuickDraw	4-97
Application-Defined Routine	4-101
Resources	4-102
The Pixel Pattern Resource	4-103
The Color Table Resource	4-104
The Color Icon Resource	4-105
Summary of Color QuickDraw	4-107
Pascal Summary	4-107
Constants	4-107
Data Types	4-109
Color QuickDraw Routines	4-113
Application-Defined Routine	4-115

C Summary	4-115
Constants	4-115
Data Types	4-118
Color QuickDraw Functions	4-122
Application-Defined Function	4-124
Assembly-Language Summary	4-124
Data Structures	4-124
Result Codes	4-128

---

Chapter 5	<b>Graphics Devices</b>	5-1
-----------	-------------------------	-----

---

About Graphics Devices	5-3
Using Graphics Devices	5-6
Optimizing Your Images for Different Graphics Devices	5-8
Zooming Windows on Multiscreen Systems	5-9
Setting a Device's Pixel Depth	5-13
Exceptional Cases When Working With Color Devices	5-13
Graphics Devices Reference	5-14
Data Structures	5-15
Routines for Graphics Devices	5-19
Creating, Setting, and Disposing of GDevice Records	5-19
Getting the Available Graphics Devices	5-25
Determining the Characteristics of a Video Device	5-29
Changing the Pixel Depth for a Video Device	5-33
Application-Defined Routine	5-35
Resource	5-37
The Screen Resource	5-37
Summary of Graphics Devices	5-38
Pascal Summary	5-38
Constants	5-38
Data Types	5-39
Routines for Graphics Devices	5-40
Application-Defined Routine	5-40
C Summary	5-41
Constants	5-41
Data Types	5-41
Functions for Graphics Devices	5-43
Application-Defined Function	5-44
Assembly-Language Summary	5-44
Data Structure	5-44
Global Variables	5-44

---

About Offscreen Graphics Worlds	6-3
Using Offscreen Graphics Worlds	6-4
Creating an Offscreen Graphics World	6-5
Setting the Graphics Port for an Offscreen Graphics World	6-8
Drawing Into an Offscreen Graphics World	6-8
Copying an Offscreen Image Into a Window	6-9
Updating an Offscreen Graphics World	6-9
Creating a Mask and a Source Image in Offscreen Graphics Worlds	6-10
Offscreen Graphics Worlds Reference	6-12
Data Structures	6-12
Routines	6-16
Creating, Altering, and Disposing of Offscreen Graphics Worlds	6-16
Saving and Restoring Graphics Ports and Offscreen Graphics Worlds	6-27
Managing an Offscreen Graphics World's Pixel Image	6-30
Summary of Offscreen Graphics Worlds	6-40
Pascal Summary	6-40
Constants	6-40
Data Types	6-41
Routines	6-42
C Summary	6-43
Constants	6-43
Data Types	6-44
Functions	6-45
Assembly-Language Summary	6-46
Result Codes	6-46

---

About Pictures	7-4
Picture Formats	7-5
Opcodes: Drawing Commands and Picture Comments	7-6
Color Pictures in Basic Graphics Ports	7-6
'PICT' Files, 'PICT' Resources, and the 'PICT' Scrap Format	7-7
The Picture Utilities	7-8
Using Pictures	7-8
Creating and Drawing Pictures	7-10
Opening and Drawing Pictures	7-13
Drawing a Picture Stored in a 'PICT' File	7-13
Drawing a Picture Stored in the Scrap	7-17
Defining a Destination Rectangle	7-18
Drawing a Picture Stored in a 'PICT' Resource	7-20
Saving Pictures	7-21
Gathering Picture Information	7-24



Pictures Reference	7-26
Data Structures	7-27
QuickDraw and Picture Utilities Routines	7-36
Creating and Disposing of Pictures	7-36
Drawing Pictures	7-43
Collecting Picture Information	7-46
Application-Defined Routines	7-61
Resources	7-67
The Picture Resource	7-67
The Color-Picking Method Resource	7-68
Summary of Pictures and the Picture Utilities	7-69
Pascal Summary	7-69
Constants	7-69
Data Types	7-69
Routines	7-72
Application-Defined Routines	7-73
C Summary	7-73
Constants	7-73
Data Types	7-74
Functions	7-76
Application-Defined Functions	7-77
Assembly-Language Summary	7-78
Data Structures	7-78
Trap Macros	7-80
Result Codes	7-80

## Chapter 8

## Cursor Utilities 8-1

---

About the Cursor	8-3
Using the Cursor Utilities	8-5
Initializing the Cursor	8-6
Changing the Appearance of the Cursor	8-7
Creating an Animated Cursor	8-13
Cursor Utilities Reference	8-16
Data Structures	8-16
Routines	8-21
Initializing Cursors	8-21
Changing Black-and-White Cursors	8-24
Changing Color Cursors	8-25
Hiding and Showing Cursors	8-28
Displaying Animated Cursors	8-31
Resources	8-33
The Cursor Resource	8-33
The Color Cursor Resource	8-34
The Animated Cursor Resource	8-36

Summary of Cursor Utilities	8-38
Pascal Summary	8-38
Constants	8-38
Data Types	8-38
Routines	8-39
C Summary	8-40
Constants	8-40
Data Types	8-41
Functions	8-42
Assembly-Language Summary	8-43
Data Structures	8-43
Global Variables	8-43

## Chapter 9      Printing Manager      9-1

---

About the Printing Manager	9-3
The Printing Graphics Port	9-4
Getting Printing Preferences From the User	9-5
QuickDraw and PostScript Printer Drivers	9-8
Page and Paper Rectangles	9-10
Printer Resolution	9-11
The TPrint Record and the Printing Loop	9-11
Print Status Dialog Boxes	9-13
Using the Printing Manager	9-15
Creating and Using a TPrint Record	9-17
Printing a Document	9-18
Printing From the Finder	9-25
Providing Names of Documents Being Printed	9-27
Printing Hints	9-27
Getting and Setting Printer Information	9-28
Determining and Setting the Resolution of the Current Printer	9-30
Determining Page Orientation	9-32
Enhancing Draft-Quality Printing	9-33
Altering the Style or Job Dialog Box	9-35
Writing an Idle Procedure	9-38
Handling Printing Errors	9-41
Printing Manager Reference	9-43
Data Structures	9-44
Printing Manager Routines	9-57
Opening and Closing the Printing Manager	9-57
Initializing and Validating TPrint Records	9-58
Displaying and Customizing the Print Dialog Boxes	9-61
Printing a Document	9-66
Optimizing Printing	9-72

Handling Printing Errors	9-75
Low-Level Routines	9-78
Application-Defined Routines	9-84
Summary of the Printing Manager	9-87
Pascal Summary	9-87
Constants	9-87
Data Types	9-88
Printing Manager Routines	9-92
Application-Defined Routines	9-93
C Summary	9-94
Constants	9-94
Data Types	9-95
Printing Manager Functions	9-99
Application-Defined Functions	9-101
Assembly-Language Summary	9-101
Data Structures	9-101
Trap Macros	9-103
Global Variable	9-103
Result Codes	9-104

---

Appendix A      **Picture Opcodes**      A-1

---

Version and Header Opcodes	A-3
Picture Opcode Data Types	A-4
Opcodes in Pictures	A-5
A Sample Extended Version 2 Picture	A-22
A Sample Version 2 Picture	A-24
A Sample Version 1 Picture	A-25

---

Appendix B      **Using Picture Comments for Printing**      B-1

---

About Picture Comments	B-3
Maintaining Device Independence	B-8
Synchronizing QuickDraw and PostScript Printer Drivers	B-10
Using Text Picture Comments	B-11
Disabling and Reenabling Line Layout	B-11
Delimiting Strings	B-16
Rotating Text	B-17
Using Graphics Picture Comments	B-22
Drawing Polygons	B-23
Rotating Graphics	B-29
Using Line-Drawing Picture Comments	B-33
Drawing Dashed Lines	B-33
Using Fractional Line Widths	B-35

Using PostScript Picture Comments	B-38
Calling PostScript Routines Directly	B-38
Optimizing PostScript Printing	B-39
Picture Comments to Avoid	B-40
Including Constants and Data Types for Picture Comments	B-42

Glossary	GL-1
----------	------

---

Index	IN-1
-------	------

---

# Figures, Tables, and Listings

## Color Plates

---

*Color plates are immediately preceding the title page.*

- Color Plate 1**     The colors of the default color tables
- Color Plate 2**     Using `CopyBits` to colorize an image
- Color Plate 3**     Examples of the `CopyMask` procedure
- Color Plate 4**     Using a mask

## Chapter 1

### Introduction to QuickDraw     1-1

---

- Figure 1-1**     A grayscale image representing bits in memory     1-6
- Figure 1-2**     The QuickDraw global coordinate plane     1-7
- Figure 1-3**     A window's local and global coordinate systems     1-8
- Figure 1-4**     The coordinate plane     1-8
- Figure 1-5**     Points and pixels     1-9
- Figure 1-6**     Drawing a line     1-10
- Figure 1-7**     Lines drawn with different bit patterns and pen sizes     1-12
- Figure 1-8**     A rectangle     1-12
- Figure 1-9**     An oval     1-13
- Figure 1-10**     An arc and a wedge     1-14
- Figure 1-11**     A rounded rectangle     1-15
- Figure 1-12**     A polygon     1-15
- Figure 1-13**     Two regions     1-16
- Figure 1-14**     A simple QuickDraw picture     1-16
- Figure 1-15**     Filling and framing various shapes     1-17
- Figure 1-16**     A two-screen system     1-21
- Figure 1-17**     The `GDevice` record and pixel map for a 4-bit video card     1-22
- Figure 1-18**     The indexed-pixel path     1-24
- Figure 1-19**     The direct-pixel path     1-25
- Figure 1-20**     The job dialog box for a StyleWriter printer     1-26
- Figure 1-21**     The job dialog box for a LaserWriter printer     1-27

## Chapter 2

### Basic QuickDraw     2-1

---

- Figure 2-1**     The `GrafPort` record and the `BitMap` record     2-8
- Figure 2-2**     A bit image     2-9
- Figure 2-3**     Relationship of the boundary rectangle and the port rectangle to the global coordinate system     2-10
- Figure 2-4**     Comparing the boundary rectangle, port rectangle, visible region, and clipping region     2-12
- Figure 2-5**     Moving a document relative to its window     2-21
- Figure 2-6**     Updating the contents of a scrolled window     2-24

<b>Figure 2-7</b>	Restoring the window origin of the port rectangle to a horizontal coordinate of 0 and a vertical coordinate of 0	2-25
<b>Figure 2-8</b>	Scrolling the image in a rectangle by using the <code>ScrollRect</code> procedure	2-44
<b>Table 2-1</b>	QuickDraw global variables	2-36
<b>Table 2-2</b>	Initial values of a basic graphics port	2-38
<b>Listing 2-1</b>	Initializing QuickDraw	2-16
<b>Listing 2-2</b>	Using the Window Manager to create a basic graphics port	2-17
<b>Listing 2-3</b>	Saving and restoring a graphics port	2-18
<b>Listing 2-4</b>	Changing global coordinates to local coordinates	2-19
<b>Listing 2-5</b>	Using <code>ScrollRect</code> to scroll the bits displayed in the window	2-22

## Chapter 3

### QuickDraw Drawing 3-1

---

<b>Figure 3-1</b>	A graphics pen	3-4
<b>Figure 3-2</b>	A bit pattern	3-5
<b>Figure 3-3</b>	Windows filled with the predefined bit patterns	3-7
<b>Figure 3-4</b>	Examples of Boolean transfer modes	3-10
<b>Figure 3-5</b>	Using the <code>LineTo</code> procedure	3-17
<b>Figure 3-6</b>	Drawing lines	3-18
<b>Figure 3-7</b>	Using the <code>LineTo</code> and <code>Line</code> procedures	3-19
<b>Figure 3-8</b>	Resizing the pen	3-19
<b>Figure 3-9</b>	Changing the pen pattern	3-21
<b>Figure 3-10</b>	Two ways to specify a rectangle	3-22
<b>Figure 3-11</b>	Drawing rectangles	3-22
<b>Figure 3-12</b>	Painting and filling rectangles	3-23
<b>Figure 3-13</b>	Drawing ovals	3-25
<b>Figure 3-14</b>	Drawing an arc and a wedge	3-26
<b>Figure 3-15</b>	A shape created by a region	3-28
<b>Figure 3-16</b>	Filling a clipping region	3-30
<b>Figure 3-17</b>	Shrinking images between graphics ports	3-33
<b>Figure 3-18</b>	Forty-five degree angles as returned by the <code>PtToAngle</code> procedure	3-57
<b>Figure 3-19</b>	Oval width and height in rounded rectangles	3-63
<b>Figure 3-20</b>	Using angles to define the radii for arcs and wedges	3-72
<b>Figure 3-21</b>	Using <code>PaintArc</code> to paint a 45° angle	3-74
<b>Figure 3-22</b>	Framing and painting polygons	3-82
<b>Figure 3-23</b>	Using <code>ScalePt</code> and <code>MapPt</code>	3-105
<b>Figure 3-24</b>	A source image and its resulting mask produced by the <code>SeedFill</code> procedure	3-109
<b>Figure 3-25</b>	Parameters for the <code>SeedFill</code> and <code>CalcMask</code> procedures	3-110
<b>Figure 3-26</b>	A source image and the resulting mask produced by the <code>CalcMask</code> procedure	3-111
<b>Figure 3-27</b>	Using <code>CopyBits</code> to stretch an image	3-113
<b>Figure 3-28</b>	Standard patterns	3-128
<b>Figure 3-29</b>	Format of a compiled pattern ('PAT ') resource	3-140
<b>Figure 3-30</b>	Format of a compiled pattern list ('PAT# ') resource	3-141

<b>Table 3-1</b>	Effect of Boolean transfer modes on 1-bit pixels	3-9
<b>Table 3-2</b>	The global variables for five predefined bit patterns	3-20
<b>Table 3-3</b>	QuickDraw routines for calculating and manipulating rectangles	3-31
<b>Table 3-4</b>	QuickDraw routines for calculating and manipulating regions	3-32
<b>Listing 3-1</b>	Drawing lines with the <code>LineTo</code> and <code>Line</code> procedures	3-18
<b>Listing 3-2</b>	Using the <code>PenSize</code> procedure	3-20
<b>Listing 3-3</b>	Using the <code>PenPat</code> procedure to change the pattern of the graphics pen	3-21
<b>Listing 3-4</b>	Using the <code>FrameRect</code> procedure to draw rectangles	3-23
<b>Listing 3-5</b>	Using the <code>PaintRect</code> and <code>FillRect</code> procedures	3-24
<b>Listing 3-6</b>	Using the <code>FrameOval</code> procedure to draw ovals	3-25
<b>Listing 3-7</b>	Using the <code>FrameArc</code> and <code>PaintArc</code> procedures	3-26
<b>Listing 3-8</b>	Creating and drawing a region	3-28
<b>Listing 3-9</b>	Creating a clipping region and filling it with a pattern	3-29
<b>Listing 3-10</b>	Creating a triangular polygon	3-30
<b>Listing 3-11</b>	Using the <code>CopyBits</code> procedure to copy between two windows	3-33

## Chapter 4

### Color QuickDraw 4-1

---

<b>Figure 4-1</b>	The color graphics port	4-7
<b>Figure 4-2</b>	The pixel map	4-10
<b>Figure 4-3</b>	Translating a 48-bit <code>RGBColor</code> record to an 8-bit pixel value on an indexed device	4-14
<b>Figure 4-4</b>	Translating an 8-bit pixel value on an indexed device to a 48-bit <code>RGBColor</code> record	4-15
<b>Figure 4-5</b>	Translating a 48-bit <code>RGBColor</code> record to a 32-bit pixel value on a direct device	4-15
<b>Figure 4-6</b>	Translating a 48-bit <code>RGBColor</code> record to a 16-bit pixel value on a direct device	4-16
<b>Figure 4-7</b>	Translating a 32-bit pixel value to a 48-bit <code>RGBColor</code> record	4-16
<b>Figure 4-8</b>	Translating a 16-bit pixel value to a 48-bit <code>RGBColor</code> record	4-17
<b>Figure 4-9</b>	Drawing with two different foreground colors (on a grayscale screen)	4-23
<b>Figure 4-10</b>	Using <code>ResEdit</code> to create a pixel pattern resource	4-24
<b>Figure 4-11</b>	Painting and filling rectangles with pixel patterns	4-25
<b>Figure 4-12</b>	Copying pixel images with the <code>CopyBits</code> procedure	4-27
<b>Figure 4-13</b>	Copying pixel images with the <code>CopyMask</code> procedure	4-29
<b>Figure 4-14</b>	Copying pixel images with the <code>CopyDeepMask</code> procedure	4-31
<b>Figure 4-15</b>	Difference between highlighting and inverting	4-42
<b>Figure 4-16</b>	Format of a compiled pixel pattern (' <code>ppat</code> ') resource	4-103
<b>Figure 4-17</b>	Format of a compiled color table (' <code>clut</code> ') resource	4-104
<b>Figure 4-18</b>	Format of a compiled color icon (' <code>cicn</code> ') resource	4-106
<b>Table 4-1</b>	Boolean source modes with colored pixels	4-33
<b>Table 4-2</b>	Arithmetic modes in a 1-bit environment	4-41
<b>Table 4-3</b>	Initial values in the <code>CGrafPort</code> record	4-64

<b>Table 4-4</b>	The colors defined by the global variable <code>QDColors</code>	4-71
<b>Table 4-5</b>	The default color tables for grayscale graphics devices	4-92
<b>Table 4-6</b>	The default color tables for color graphics devices	4-93
<b>Listing 4-1</b>	Using the Window Manager to create a color graphics port	4-20
<b>Listing 4-2</b>	Changing the foreground color	4-22
<b>Listing 4-3</b>	Rez input for a pixel pattern resource	4-24
<b>Listing 4-4</b>	Using pixel patterns to paint and fill	4-25
<b>Listing 4-5</b>	Using <code>CopyBits</code> to produce coloration effects	4-35
<b>Listing 4-6</b>	Setting the highlight bit	4-42
<b>Listing 4-7</b>	Using highlighting for text	4-43

## Chapter 5

### Graphics Devices 5-1

---

<b>Figure 5-1</b>	The <code>GDevice</code> record	5-5
<b>Listing 5-1</b>	Using the <code>DeviceLoop</code> procedure	5-8
<b>Listing 5-2</b>	Drawing into different screens	5-9
<b>Listing 5-3</b>	Zooming a window	5-10

## Chapter 6

### Offscreen Graphics Worlds 6-1

---

<b>Listing 6-1</b>	Using a single offscreen graphics world and the <code>CopyBits</code> procedure	6-5
<b>Listing 6-2</b>	Using two offscreen graphics worlds and the <code>CopyMask</code> procedure	6-10

## Chapter 7

### Pictures 7-1

---

<b>Figure 7-1</b>	A picture of a party hat	7-4
<b>Figure 7-2</b>	The <code>Picture</code> record	7-5
<b>Figure 7-3</b>	A simple picture	7-12
<b>Figure 7-4</b>	Structure of a compiled picture ( <code>'PICT'</code> ) resource	7-68
<b>Table 7-1</b>	Routine selectors for an application-defined color-picking method	7-61
<b>Listing 7-1</b>	Creating and drawing a picture	7-11
<b>Listing 7-2</b>	Opening and drawing a picture from disk	7-13
<b>Listing 7-3</b>	Replacing QuickDraw's standard low-level picture-reading routine	7-15
<b>Listing 7-4</b>	Determining whether a graphics port is color or basic	7-16
<b>Listing 7-5</b>	A custom low-level procedure for spooling a picture from disk	7-16
<b>Listing 7-6</b>	Pasting in a picture from the scrap	7-17
<b>Listing 7-7</b>	Adjusting the destination rectangle for a picture	7-18
<b>Listing 7-8</b>	Drawing a picture stored in a resource file	7-20
<b>Listing 7-9</b>	Saving a picture as a <code>'PICT'</code> file	7-21
<b>Listing 7-10</b>	Replacing QuickDraw's standard low-level picture-writing routine	7-22



<b>Listing 7-11</b>	A custom low-level routine for spooling a picture to disk	7-23
<b>Listing 7-12</b>	Looking for color profile comments in a picture	7-25

## Chapter 8

### Cursor Utilities 8-1

---

<b>Figure 8-1</b>	Hot spots in cursors	8-4
<b>Figure 8-2</b>	The standard arrow cursor	8-8
<b>Figure 8-3</b>	The I-beam, crosshairs, plus sign, and wristwatch cursors	8-8
<b>Figure 8-4</b>	A window and its arrow and I-beam regions	8-9
<b>Figure 8-5</b>	Changing the cursor from the I-beam cursor to the arrow cursor	8-10
<b>Figure 8-6</b>	The 'CURS' resources for an animated globe cursor	8-13
<b>Figure 8-7</b>	An 'acur' resource for an animated cursor	8-14
<b>Figure 8-8</b>	Format of a compiled cursor ('CURS') resource	8-34
<b>Figure 8-9</b>	Format of a compiled color cursor ('crsr') resource	8-35
<b>Figure 8-10</b>	Format of a compiled animated cursor ('acur') resource	8-37
<b>Table 8-1</b>	Cursor appearance	8-17
<b>Listing 8-1</b>	Initializing the Cursor Utilities	8-6
<b>Listing 8-2</b>	Changing the cursor	8-10
<b>Listing 8-3</b>	Animating a cursor with the <code>RotateCursor</code> procedure	8-15
<b>Listing 8-4</b>	Animating a cursor with the <code>SpinCursor</code> procedure	8-15

## Chapter 9

### Printing Manager 9-1

---

<b>Figure 9-1</b>	A standard File menu for an application	9-5
<b>Figure 9-2</b>	The style dialog box for a StyleWriter printer	9-6
<b>Figure 9-3</b>	The style dialog box for a LaserWriter printer	9-7
<b>Figure 9-4</b>	The job dialog box for a StyleWriter printer	9-7
<b>Figure 9-5</b>	The job dialog box for a LaserWriter printer	9-8
<b>Figure 9-6</b>	Page and paper rectangles	9-10
<b>Figure 9-7</b>	A <code>TPrint</code> record	9-12
<b>Figure 9-8</b>	The print status dialog box for a LaserWriter printer driver printing in the background	9-13
<b>Figure 9-9</b>	A status dialog box with the LaserWriter printer driver's print status dialog box	9-14
<b>Figure 9-10</b>	How the <code>PrJobMerge</code> procedure works	9-26
<b>Figure 9-11</b>	Sample resolutions for a PostScript printer and a QuickDraw printer	9-30
<b>Figure 9-12</b>	A print job dialog box with additional checkboxes	9-35
<b>Table 9-1</b>	Values for the <code>lParam1</code> parameter when using the <code>iPrDevCtl</code> control constant	9-83
<b>Listing 9-1</b>	Reading a document's <code>TPrint</code> record	9-17
<b>Listing 9-2</b>	A sample printing loop	9-20
<b>Listing 9-3</b>	Checking whether the current printer driver supports the <code>PrGeneral</code> procedure	9-29

<b>Listing 9-4</b>	Using the <code>getRslDataOp</code> and <code>setRslOp</code> opcodes with the <code>PrGeneral</code> procedure 9-31
<b>Listing 9-5</b>	Using the <code>getRotnOp</code> opcode with the <code>PrGeneral</code> procedure to determine page orientation 9-33
<b>Listing 9-6</b>	Using the <code>draftBitsOp</code> opcode with the <code>PrGeneral</code> procedure for enhanced draft-quality printing 9-34
<b>Listing 9-7</b>	Installing an initialization function to alter the print job dialog box 9-37
<b>Listing 9-8</b>	Adding items to a print job dialog box 9-37
<b>Listing 9-9</b>	An idle procedure 9-40

## Appendix A

### Picture Opcodes A-1

---

<b>Figure A-1</b>	A picture A-23
<b>Table A-1</b>	Data types for picture opcodes A-4
<b>Table A-2</b>	Opcodes for extended version 2 and version 2 pictures A-5
<b>Table A-3</b>	Opcodes for version 1 pictures A-18
<b>Listing A-1</b>	Data for the <code>BkPixPat</code> , <code>PnPixPat</code> , and <code>FillPixPat</code> opcodes A-17
<b>Listing A-2</b>	Data for the <code>BitsRect</code> and <code>PackBitsRect</code> opcodes A-17
<b>Listing A-3</b>	Data for the <code>BitsRgn</code> and <code>PackBitsRgn</code> opcodes A-18
<b>Listing A-4</b>	Creating and drawing an extended version 2 picture A-22
<b>Listing A-5</b>	A decompiled extended version 2 picture A-23
<b>Listing A-6</b>	A decompiled version 2 picture A-24
<b>Listing A-7</b>	A decompiled version 1 picture A-25

## Appendix B

### Using Picture Comments for Printing B-1

---

<b>Figure B-1</b>	The line layout error between a bitmapped font and a PostScript font B-12
<b>Figure B-2</b>	Major and minor glyphs B-13
<b>Figure B-3</b>	Distributing layout error to the major glyphs B-13
<b>Figure B-4</b>	Distributing layout error among major and minor glyphs B-14
<b>Figure B-5</b>	Using the <code>LineLayoutOff</code> and <code>LineLayoutOn</code> picture comments B-15
<b>Figure B-6</b>	Variations in text alignment B-18
<b>Figure B-7</b>	Types of polygons B-23
<b>Figure B-8</b>	QuickDraw and PostScript polygons B-29
<b>Figure B-9</b>	Changing the pen width using the <code>SetLineWidth</code> picture comment B-36
<b>Table B-1</b>	Names, values, and data sizes for picture comments B-5
<b>Table B-2</b>	Low-level QuickDraw routines disabled by the <code>PostScriptBegin</code> comment B-9
<b>Listing B-1</b>	Synchronizing QuickDraw and the PostScript driver B-10
<b>Listing B-2</b>	Flushing the buffer for a PostScript printer driver B-11
<b>Listing B-3</b>	Disabling line layout by using the <code>LineLayoutOff</code> and <code>StringBegin</code> picture comments B-17

<b>Listing B-4</b>	Displaying rotated text using picture comments	B-21
<b>Listing B-5</b>	Creating polygons	B-26
<b>Listing B-6</b>	Drawing polygons	B-27
<b>Listing B-7</b>	Using picture comments to rotate graphics	B-31
<b>Listing B-8</b>	Using the <code>RotateCenter</code> , <code>RotateBegin</code> , and <code>RotateEnd</code> picture comments	B-32
<b>Listing B-9</b>	Using the <code>DashedLine</code> picture comment	B-34
<b>Listing B-10</b>	Using the <code>SetLineWidth</code> picture comment	B-37
<b>Listing B-11</b>	Sending PostScript code directly to the printer	B-39



# About This Book

---

This book, *Inside Macintosh: Imaging With QuickDraw*, describes how to create images, display them in black and white or color, and print them using QuickDraw—the imaging engine available on all Macintosh computers. The chapters in this book and the information they contain are summarized here.

- “Introduction to QuickDraw” introduces you to the terms, concepts, and capabilities of QuickDraw.
- “Basic QuickDraw” describes how to create and manage the basic graphics port—the drawing environment in which your application can create graphics and text in either black and white or eight basic colors.
- “QuickDraw Drawing” explains the routines and data structures—common to both basic QuickDraw and Color QuickDraw—that your application can use to draw lines, rectangles, rounded rectangles, ovals, arcs, wedges, polygons, and regions, and to copy images from one graphics port to another. This chapter also describes the routines that you can use to perform calculations and other manipulations of these shapes—including comparing them and finding their unions and intersections.
- “Color QuickDraw” describes the version of QuickDraw that provides a range of color and grayscale capabilities to your application. You should read this chapter if your application needs to use shades of gray or more colors than the eight predefined colors provided by basic QuickDraw.
- “Graphics Devices” describes how Color QuickDraw manages video devices so that your application can draw to a window’s graphics port without regard to the capabilities of the screen—even if the window spans more than one screen.
- “Offscreen Graphics Worlds” describes how you can improve your application’s appearance and performance by constructing images in offscreen graphics worlds before drawing them onscreen.
- “Pictures” describes how to create and draw QuickDraw pictures, which are sequences of saved drawing commands that your application can share among documents, even among documents created by other applications.
- “Cursor Utilities” describes how to create and use cursors, including color and animated cursors, for indicating the relative mouse location onscreen and for indicating when your application is performing a lengthy task.
- “Printing Manager” describes how your application can print by creating a printing graphics port and drawing into it using QuickDraw routines. This chapter also explains how to implement the user interface that helps users when they wish to print.

## PREFACE

- Appendix A, “Picture Opcodes,” describes opcodes used by the `DrawPicture` procedure to determine what object to draw or what mode to change for subsequent drawing. Your application generally should not read or write picture opcodes directly but should instead use the `QuickDraw` routines described in the chapter “Pictures” for generating and processing the opcodes. Picture opcodes are listed in this appendix for your application’s debugging purposes.
- Appendix B, “Using Picture Comments for Printing,” describes how your application can use picture comments to instruct printer drivers to perform graphics operations that `QuickDraw` does not support.

Most applications draw in windows, which are rectangular areas that are usually subsets of the screen. Each window represents its own `QuickDraw` graphics port. When you create a window, the Window Manager uses `QuickDraw` to create a graphics port in which the window’s contents are displayed. This book describes how to draw inside a window. See the chapter “Window Manager” in *Inside Macintosh: Macintosh Toolbox Essentials* for a complete description of creating and managing the windows inside which your application draws.

In addition to the features described in this book, `QuickDraw` also provides extensive text support. However, most of that support is not described in this book; for information on handling text in your application, see instead *Inside Macintosh: Text*.

If you are new to programming for the Macintosh computer, you should also read *Inside Macintosh: Overview* for an introduction to general concepts of Macintosh programming.

## Format of a Typical Chapter

---

The chapters in this book follow a standard structure. For example, the chapter “Color `QuickDraw`” contains these sections:

- “About Color `QuickDraw`.” This section introduces the drawing environment provided by Color `QuickDraw`, the data structure with which your application specifies colors to Color `QuickDraw`, and the manner in which Color `QuickDraw` translates the colors that your application specifies into colors on the user’s screen.
- “Using Color `QuickDraw`.” Using code samples and step-by-step instructions, this section describes how to use Color `QuickDraw` to accomplish the basic tasks necessary for drawing in color.

## PREFACE

- “Color QuickDraw Reference.” This section provides a complete reference to the data structures, routines, and resources that your application can use to draw with Color QuickDraw. 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 Color QuickDraw.” This section provides the Pascal and C interfaces for the constants, data structures, routines, and result codes associated with Color QuickDraw drawing. 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 uses special formats so that you can scan it quickly.

### Special Fonts

---

All code listings, reserved words, and 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 2-3 in the chapter “Basic QuickDraw.”) ◆

#### IMPORTANT

A note like this contains information that is essential for an understanding of the main text. (An example appears on page 2-4 in the chapter “Basic QuickDraw.”) ▲

#### ▲ 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 2-31 in the chapter “Basic QuickDraw.”) ▲

## PREFACE

### Empty Strings

---

This book occasionally instructs you to provide an empty string in routine parameters and resources. How you specify an empty string depends on what language and development environment you are using. In Rez input files and in C code, for example, you specify an empty string by using two double quotation marks (""), and in Pascal you specify an empty string by using two single quotation marks ('').

### Assembly-Language Information

---

*Inside Macintosh* provides information about the trap macro and routine selector for specific routines like this:

Trap macro	Selector
<code>_QDExtensions</code>	<code>\$00E0010</code>

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

0	<code>baseAddr</code>	long	bitmap base address
4	<code>rowBytes</code>	word	row bytes (must be even)
6	<code>bounds</code>	8 bytes	boundary rectangle

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 assembly-language 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 “Data Structures” in the main text 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



## PREFACE

the list of figures, tables, and listings. If you know the name of a particular routine (such as `MyDrawLines`) 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.

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.

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 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 any additional technical information, contact

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

