

Figures, Tables, and Listings

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 <code>GDevice</code> 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 <code>GrafPort</code> record and the <code>BitMap</code> 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
Listing 2-1	Initializing QuickDraw	2-16
Listing 2-2	Using the Window Manager to create a basic graphics port	2-17
Listing 2-3	Saving and restoring a graphics port	2-18
Listing 2-4	Changing global coordinates to local coordinates	2-19
Figure 2-5	Moving a document relative to its window	2-21
Listing 2-5	Using <code>ScrollRect</code> to scroll the bits displayed in the window	2-22
Figure 2-6	Updating the contents of a scrolled window	2-24
Figure 2-7	Restoring the window origin of the port rectangle to a horizontal coordinate of 0 and a vertical coordinate of 0	2-25
Table 2-1	QuickDraw global variables	2-36
Table 2-2	Initial values of a basic graphics port	2-38
Figure 2-8	Scrolling the image in a rectangle by using the <code>ScrollRect</code> procedure	2-44

Figure 3-1	A graphics pen	3-4
Figure 3-2	A bit pattern	3-5
Figure 3-3	Windows filled with the predefined bit patterns	3-7
Table 3-1	Effect of Boolean transfer modes on 1-bit pixels	3-9
Figure 3-4	Examples of Boolean transfer modes	3-10
Figure 3-5	Using the <code>LineTo</code> procedure	3-17
Listing 3-1	Drawing lines with the <code>LineTo</code> and <code>Line</code> procedures	3-18
Figure 3-6	Drawing lines	3-18
Figure 3-7	Using the <code>LineTo</code> and <code>Line</code> procedures	3-19
Figure 3-8	Resizing the pen	3-19
Listing 3-2	Using the <code>PenSize</code> procedure	3-20
Table 3-2	The global variables for five predefined bit patterns	3-20
Figure 3-9	Changing the pen pattern	3-21
Listing 3-3	Using the <code>PenPat</code> procedure to change the pattern of the graphics pen	3-21
Figure 3-10	Two ways to specify a rectangle	3-22
Figure 3-11	Drawing rectangles	3-22
Listing 3-4	Using the <code>FrameRect</code> procedure to draw rectangles	3-23
Figure 3-12	Painting and filling rectangles	3-23
Listing 3-5	Using the <code>PaintRect</code> and <code>FillRect</code> procedures	3-24
Figure 3-13	Drawing ovals	3-25
Listing 3-6	Using the <code>FrameOval</code> procedure to draw ovals	3-25
Figure 3-14	Drawing an arc and a wedge	3-26
Listing 3-7	Using the <code>FrameArc</code> and <code>PaintArc</code> procedures	3-26
Listing 3-8	Creating and drawing a region	3-28
Figure 3-15	A shape created by a region	3-28
Listing 3-9	Creating a clipping region and filling it with a pattern	3-29
Figure 3-16	Filling a clipping region	3-30
Listing 3-10	Creating a triangular polygon	3-30
Table 3-3	QuickDraw routines for calculating and manipulating rectangles	3-31
Table 3-4	QuickDraw routines for calculating and manipulating regions	3-32
Figure 3-17	Shrinking images between graphics ports	3-33
Listing 3-11	Using the <code>CopyBits</code> procedure to copy between two windows	3-33
Figure 3-18	Forty-five-degree angles as returned by the <code>PtToAngle</code> procedure	3-57
Figure 3-19	Oval width and height in rounded rectangles	3-63
Figure 3-20	Using angles to define the radii for arcs and wedges	3-72
Figure 3-21	Using <code>PaintArc</code> to paint a 45° angle	3-74
Figure 3-22	Framing and painting polygons	3-82
Figure 3-23	Using <code>ScalePt</code> and <code>MapPt</code>	3-105
Figure 3-24	A source image and its resulting mask produced by the <code>SeedFill</code> procedure	3-109
Figure 3-25	Parameters for the <code>SeedFill</code> and <code>CalcMask</code> procedures	3-110
Figure 3-26	A source image and the resulting mask produced by the <code>CalcMask</code> procedure	3-111
Figure 3-27	Using <code>CopyBits</code> to stretch an image	3-113

Figure 3-28	Standard patterns	3-128
Figure 3-29	Format of a compiled pattern ('PAT') resource	3-140
Figure 3-30	Format of a compiled pattern list ('PAT#') resource	3-141

Chapter 4

Color QuickDraw 4-1

Figure 4-1	The color graphics port	4-7
Figure 4-2	The pixel map	4-10
Figure 4-3	Translating a 48-bit <code>RGBColor</code> record to an 8-bit pixel value on an indexed device	4-14
Figure 4-4	Translating an 8-bit pixel value on an indexed device to a 48-bit <code>RGBColor</code> record	4-15
Figure 4-5	Translating a 48-bit <code>RGBColor</code> record to a 32-bit pixel value on a direct device	4-15
Figure 4-6	Translating a 48-bit <code>RGBColor</code> record to a 16-bit pixel value on a direct device	4-16
Figure 4-7	Translating a 32-bit pixel value to a 48-bit <code>RGBColor</code> record	4-16
Figure 4-8	Translating a 16-bit pixel value to a 48-bit <code>RGBColor</code> record	4-17
Listing 4-1	Using the Window Manager to create a color graphics port	4-20
Listing 4-2	Changing the foreground color	4-22
Figure 4-9	Drawing with two different foreground colors (on a grayscale screen)	4-23
Figure 4-10	Using ResEdit to create a pixel pattern resource	4-24
Listing 4-3	Rez input for a pixel pattern resource	4-24
Listing 4-4	Using pixel patterns to paint and fill	4-25
Figure 4-11	Painting and filling rectangles with pixel patterns	4-25
Figure 4-12	Copying pixel images with the <code>CopyBits</code> procedure	4-27
Figure 4-13	Copying pixel images with the <code>CopyMask</code> procedure	4-29
Figure 4-14	Copying pixel images with the <code>CopyDeepMask</code> procedure	4-31
Table 4-1	Boolean source modes with colored pixels	4-33
Listing 4-5	Using <code>CopyBits</code> to produce coloration effects	4-35
Table 4-2	Arithmetic modes in a 1-bit environment	4-41
Figure 4-15	Difference between highlighting and inverting	4-42
Listing 4-6	Setting the highlight bit	4-42
Listing 4-7	Using highlighting for text	4-43
Table 4-3	Initial values in the <code>CGrafPort</code> record	4-64
Table 4-4	The colors defined by the global variable <code>QDColors</code>	4-71
Table 4-5	The default color tables for grayscale graphics devices	4-92
Table 4-6	The default color tables for color graphics devices	4-93
Figure 4-16	Format of a compiled pixel pattern ('ppat') resource	4-103
Figure 4-17	Format of a compiled color table ('clut') resource	4-104
Figure 4-18	Format of a compiled color icon ('cicn') resource	4-106

Chapter 5	Graphics Devices	5-1
	Figure 5-1	The <code>GDevice</code> record 5-5
	Listing 5-1	Using the <code>DeviceLoop</code> procedure 5-8
	Listing 5-2	Drawing into different screens 5-9
	Listing 5-3	Zooming a window 5-10
Chapter 6	Offscreen Graphics Worlds	6-1
	Listing 6-1	Using a single offscreen graphics world and the <code>CopyBits</code> procedure 6-5
	Listing 6-2	Using two offscreen graphics worlds and the <code>CopyMask</code> procedure 6-10
Chapter 7	Pictures	7-1
	Figure 7-1	A picture of a party hat 7-4
	Figure 7-2	The <code>Picture</code> record 7-5
	Listing 7-1	Creating and drawing a picture 7-11
	Figure 7-3	A simple picture 7-12
	Listing 7-2	Opening and drawing a picture from disk 7-13
	Listing 7-3	Replacing QuickDraw's standard low-level picture-reading routine 7-15
	Listing 7-4	Determining whether a graphics port is color or basic 7-16
	Listing 7-5	A custom low-level procedure for spooling a picture from disk 7-16
	Listing 7-6	Pasting in a picture from the scrap 7-17
	Listing 7-7	Adjusting the destination rectangle for a picture 7-18
	Listing 7-8	Drawing a picture stored in a resource file 7-20
	Listing 7-9	Saving a picture as a 'PICT' file 7-21
	Listing 7-10	Replacing QuickDraw's standard low-level picture-writing routine 7-22
	Listing 7-11	A custom low-level routine for spooling a picture to disk 7-23
	Listing 7-12	Looking for color profile comments in a picture 7-25
	Table 7-1	Routine selectors for an application-defined color-picking method 7-61
	Figure 7-4	Structure of a compiled picture ('PICT') resource 7-68
Chapter 8	Cursor Utilities	8-1
	Figure 8-1	Hot spots in cursors 8-4
	Listing 8-1	Initializing the Cursor Utilities 8-6
	Figure 8-2	The standard arrow cursor 8-8
	Figure 8-3	The I-beam, crosshairs, plus sign, and wristwatch cursors 8-8
	Figure 8-4	A window and its arrow and I-beam regions 8-9
	Figure 8-5	Changing the cursor from the I-beam cursor to the arrow cursor 8-10
	Listing 8-2	Changing the cursor 8-10

Figure 8-6	The 'CURS' resources for an animated globe cursor	8-13
Figure 8-7	An 'acur' resource for an animated cursor	8-14
Listing 8-3	Animating a cursor with the <code>RotateCursor</code> procedure	8-15
Listing 8-4	Animating a cursor with the <code>SpinCursor</code> procedure	8-15
Table 8-1	Cursor appearance	8-17
Figure 8-8	Format of a compiled cursor ('CURS') resource	8-34
Figure 8-9	Format of a compiled color cursor ('crsr') resource	8-35
Figure 8-10	Format of a compiled animated cursor ('acur') resource	8-37

Chapter 9

Printing Manager 9-1

Figure 9-1	A standard File menu for an application	9-5
Figure 9-2	The style dialog box for a StyleWriter printer	9-6
Figure 9-3	The style dialog box for a LaserWriter printer	9-7
Figure 9-4	The job dialog box for a StyleWriter printer	9-7
Figure 9-5	The job dialog box for a LaserWriter printer	9-8
Figure 9-6	Page and paper rectangles	9-10
Figure 9-7	A <code>TPrint</code> record	9-12
Figure 9-8	The print status dialog box for a LaserWriter printer driver printing in the background	9-13
Figure 9-9	A status dialog box with the LaserWriter printer driver's print status dialog box	9-14
Listing 9-1	Reading a document's <code>TPrint</code> record	9-17
Listing 9-2	A sample printing loop	9-20
Figure 9-10	How the <code>PrJobMerge</code> procedure works	9-26
Listing 9-3	Checking whether the current printer driver supports the <code>PrGeneral</code> procedure	9-29
Figure 9-11	Sample resolutions for a PostScript printer and a QuickDraw printer	9-30
Listing 9-4	Using the <code>getRslDataOp</code> and <code>setRslOp</code> opcodes with the <code>PrGeneral</code> procedure	9-31
Listing 9-5	Using the <code>getRotnOp</code> opcode with the <code>PrGeneral</code> procedure to determine page orientation	9-33
Listing 9-6	Using the <code>draftBitsOp</code> opcode with the <code>PrGeneral</code> procedure for enhanced draft-quality printing	9-34
Figure 9-12	A print job dialog box with additional checkboxes	9-35
Listing 9-7	Installing an initialization function to alter the print job dialog box	9-37
Listing 9-8	Adding items to a print job dialog box	9-37
Listing 9-9	An idle procedure	9-40
Table 9-1	Values for the <code>lParam1</code> parameter when using the <code>iPrDevCtl</code> control constant	9-83

Appendix A

Picture Opcodes A-1

Table A-1	Data types for picture opcodes	A-4
Table A-2	Opcodes for extended version 2 and version 2 pictures	A-5
Listing A-1	Data for the <code>BkPixPat</code> , <code>PnPixPat</code> , and <code>FillPixPat</code> opcodes	A-17

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

Appendix B

Using Picture Comments for Printing B-1

Table B-1	Names, values, and data sizes for picture comments	B-5
Table B-2	Low-level QuickDraw routines disabled by the <code>PostScriptBegin</code> comment	B-9
Listing B-1	Synchronizing QuickDraw and the PostScript driver	B-10
Listing B-2	Flushing the buffer for a PostScript printer driver	B-11
Figure B-1	The line layout error between a bitmapped font and a PostScript font	B-12
Figure B-2	Major and minor glyphs	B-13
Figure B-3	Distributing layout error to the major glyphs	B-13
Figure B-4	Distributing layout error among major and minor glyphs	B-14
Figure B-5	Using the <code>LineLayoutOff</code> and <code>LineLayoutOn</code> picture comments	B-15
Listing B-3	Disabling line layout by using the <code>LineLayoutOff</code> and <code>StringBegin</code> picture comments	B-17
Figure B-6	Variations in text alignment	B-18
Listing B-4	Displaying rotated text using picture comments	B-21
Figure B-7	Types of polygons	B-23
Listing B-5	Creating polygons	B-26
Listing B-6	Drawing polygons	B-27
Figure B-8	QuickDraw and PostScript polygons	B-29
Listing B-7	Using picture comments to rotate graphics	B-31
Listing B-8	Using the <code>RotateCenter</code> , <code>RotateBegin</code> , and <code>RotateEnd</code> picture comments	B-32
Listing B-9	Using the <code>DashedLine</code> picture comment	B-34
Figure B-9	Changing the pen width using the <code>SetLineWidth</code> picture comment	B-36
Listing B-10	Using the <code>SetLineWidth</code> picture comment	B-37
Listing B-11	Sending PostScript code directly to the printer	B-39

