

Figures, Tables, and Listings

Preface

About This Book xxi

Figure P-1 Roadmap to the QuickDraw GX suite of books xxii

Chapter 1

Introduction to QuickDraw GX Typography 1-1

Figure 1-1 Contextual forms in a Roman font 1-4
Figure 1-2 Elements that distinguish glyphs of a Roman font 1-5
Figure 1-3 The Standard Roman character set 1-6
Figure 1-4 Input order and display order 1-8
Figure 1-5 Terms for glyph measurements 1-9
Figure 1-6 Font variations along a variation axis 1-10
Figure 1-7 An example of an unusual text face 1-11
Figure 1-8 Multi-direction text 1-12
Figure 1-9 Leading edges and trailing edges 1-13
Figure 1-10 Baselines for different sizes of a glyph and for different writing systems 1-14
Figure 1-11 Drop capitals 1-14
Figure 1-12 Three style runs in a line of text 1-15
Figure 1-13 Two direction runs in a line of text 1-15
Figure 1-14 Contextual forms of the Arabic letter “ha” 1-16
Figure 1-15 Examples of Roman ligatures 1-16
Figure 1-16 Different kinds of alignment 1-17
Figure 1-17 Glyphs with and without kerning 1-18
Figure 1-18 Normal, tight, and loose tracking by the selection of track setting 1-19
Table 1-1 Some special font features for layout shapes 1-19
Figure 1-19 Determining where to break a line 1-20
Figure 1-20 Caret position 1-22
Figure 1-21 Highlighting 1-23
Figure 1-22 Discontiguous highlighting 1-23
Figure 1-23 Hit-testing 1-24

Chapter 2

Typographic Shapes 2-1

Figure 2-1 Text shape 2-3
Figure 2-2 Glyph shape 2-4
Figure 2-3 Layout shape 2-5
Figure 2-4 Geometry of a typographic shape 2-5
Figure 2-5 Standard bounding rectangle and typographic bounding rectangle 2-7
Figure 2-6 Hit-testing a typographic shape 2-9
Figure 2-7 Effects of the GXSetShapeBounds function 2-11
Table 2-1 Results of converting typographic shapes to other types of shapes 2-13

Table 2-2	Converting a typographic shape to another typographic shape 2-13
Table 2-3	Setting the shape parts of various types of shapes 2-15
Table 2-4	Selected effects of shape-related functions that you can apply to typographic shapes 2-17
Table 2-5	Geometric shape functions that you can apply to typographic shapes 2-17
Table 2-6	Geometric operations that you can apply to typographic shapes 2-18
Table 2-7	Selected transform-related functions that you can apply to typographic shapes 2-21

Chapter 3

Text Shapes 3-1

Figure 3-1	Geometry of a text shape 3-3
Figure 3-2	Three examples of a text shape, each with a different style applied 3-5
Listing 3-1	Creating a text shape with a nondefault text size 3-6
Listing 3-2	Replacing text in a text shape 3-7
Table 3-1	Changing text in a text shape using the <code>GXSetTextParts</code> function 3-7

Chapter 4

Glyph Shapes 4-1

Figure 4-1	Geometry of a glyph shape 4-4
Figure 4-2	The effect of the positions and advance bits arrays on glyph placement 4-5
Figure 4-3	The same shape with a new advance bits array 4-6
Figure 4-4	Various tangents 4-7
Figure 4-5	The effect of the tangents array on glyph placement 4-8
Figure 4-6	Tangents used with and without positions 4-8
Figure 4-7	The effect of style runs on the appearance of glyphs in a glyph shape 4-9
Figure 4-8	An example of a glyph shape with a style run for each glyph 4-9
Listing 4-1	Creating a glyph shape with style runs 4-11
Figure 4-9	A glyph shape with two styles 4-12
Listing 4-2	Getting all of the information from a glyph shape 4-12
Listing 4-3	Inserting text into an existing glyph shape 4-14
Table 4-1	Changing text in a glyph shape using the <code>GXSetGlyphParts</code> function 4-14
Listing 4-4	Changing the style runs of a glyph shape 4-15
Listing 4-5	Setting the positions and advance bits arrays of a glyph shape 4-17
Figure 4-10	A glyph shape with positions and advance bits arrays set 4-18
Figure 4-11	A glyph shape with 45-degree angle tangents 4-18
Listing 4-6	Creating a series of tangents with varying angles and scales 4-19
Figure 4-12	Varying the angle and scale of individual glyphs using tangents 4-21

Figure 5-1	Geometry of a layout shape	5-4
Figure 5-2	An example of a layout with its text, style, and direction-level runs marked	5-5
Figure 5-3	English, Arabic, and Japanese text directions	5-6
Figure 5-4	A line of text rotated into a vertical position	5-8
Figure 5-5	A line of right-to-left of text with multiple direction levels	5-9
Table 5-1	Interactions between the <code>width</code> , <code>just</code> , and <code>flush</code> fields	5-11
Figure 5-6	Types of alignment	5-12
Figure 5-7	Use of the <code>flush</code> field	5-12
Figure 5-8	Alignment and justification in English	5-13
Figure 5-9	Alignment and justification in Arabic	5-13
Figure 5-10	Use of the <code>just</code> field	5-14
Figure 5-11	How different values for justification and alignment affect text in a layout shape	5-15
Figure 5-12	Text with multiple baselines aligned to the default baseline	5-16
Listing 5-1	Creating and drawing a layout shape	5-18
Listing 5-2	Creating a line containing multiple style runs	5-19
Figure 5-13	A layout shape with multiple style runs	5-20
Listing 5-3	Adding text to a layout shape using the <code>GXSetLayoutParts</code> function	5-21
Table 5-2	Changing text in a layout shape using the <code>GXSetLayoutParts</code> function	5-21
Listing 5-4	Inserting a text shape and a glyph shape into a layout shape	5-22
Listing 5-5	Creating a new layout shape from a previously existing one	5-23
Listing 5-6	Altering the alignment of a layout shape	5-24
Figure 5-14	Changing the alignment of a layout shape	5-26

Figure 6-1	The style object used by all typographic shapes	6-4
Figure 6-2	Face layers combined to form the visual composite of a Roman "E"	6-6
Figure 6-3	An underlined glyph with tangent values	6-8
Figure 6-4	Underlining with interval and with style changes	6-8
Figure 6-5	Underlining vertical text through its center	6-9
Table 6-1	Layer flag values and descriptions	6-10
Table 6-2	Alignment values and descriptions	6-11
Figure 6-6	Comparing alignment values for horizontal and vertical text	6-12
Figure 6-7	Comparing alignment values for full justification	6-13
Table 6-3	Text attributes and their values	6-15
Figure 6-8	Orienting text vertically and horizontally	6-16
Listing 6-1	Advance mapping	6-18
Figure 6-9	A shape with the advance mapping applied	6-18
Listing 6-2	Creating an italic text face	6-19
Figure 6-10	An italic text face	6-20
Figure 6-11	A condensed text face	6-21
Listing 6-3	Creating a drop-shadow text face	6-21

Figure 6-12	A drop-shadow text face	6-22
Figure 6-13	Different values of boldface	6-22
Listing 6-4	Creating a simple underline text face	6-23
Figure 6-14	A simple underline text face	6-23
Listing 6-5	Creating a thicker underline	6-24
Figure 6-15	A thicker underline	6-25
Listing 6-6	Using the automatic text advance attribute	6-26
Figure 6-16	Drawing text using the <code>gxAutoAdvanceText</code> text attribute	6-27
Listing 6-7	Using the no contour grid text attribute	6-27
Figure 6-17	Turning the no contour grid attribute off and on	6-28
Listing 6-8	Setting the vertical text attribute	6-29
Listing 6-9	The effects of the vertical text attribute on a glyph shape	6-29
Figure 6-18	Using the <code>gxVerticalText</code> attribute with a text or glyph shape	6-31
Figure 6-19	Using the <code>gxVerticalText</code> attribute with a layout shape	6-31
Listing 6-10	Filling a typographic shape with a pattern	6-32
Figure 6-20	A typographic shape with a pattern	6-33
Listing 6-11	Creating an unusual effect	6-34
Figure 6-21	An unusual effect with text faces	6-35

Chapter 7

Font Objects	7-1	
Figure 7-1	The QuickDraw GX font object and its accessible properties	7-6
Figure 7-2	Words with alphabetic, syllabic, and ideographic characters	7-8
Table 7-1	Character code sizes among various platforms and scripts	7-9
Figure 7-3	Font variations along the 'wght' variation axis	7-10
Table 7-2	A list of predefined font descriptors	7-10
Figure 7-4	Font variations for the 'wght' and 'wdth' axes	7-11
Table 7-3	QuickDraw GX storage types	7-13
Table 7-4	Font tables and their contents	7-14
Listing 7-1	Obtaining a list of available fonts in the system	7-16
Listing 7-2	Using the GXGetFontName function	7-17
Listing 7-3	Extracting a full name as a C string	7-18
Listing 7-4	Adding a new font name to a font	7-18
Listing 7-5	Retrieving an array of ligature settings	7-19
Listing 7-6	Determining font variations	7-20
Listing 7-7	Retrieving all fonts that support Japanese characters	7-20
Listing 7-8	Using the GXGetFontTable function to retrieve a table	7-21

Chapter 8

Layout Styles	8-1
Figure 8-1	Layout-specific properties of the style object discussed in this chapter 8-4
Figure 8-2	Negative and positive with-stream shift 8-7
Figure 8-3	Combining with-stream and cross-stream shift 8-7
Figure 8-4	Caret position between with-stream shifted glyphs 8-7
Figure 8-5	Apparent kerning caused by a glyph that extends beyond its advance width 8-8
Figure 8-6	When kerning can and cannot occur 8-8

Figure 8-7	Caret position between two kerned glyphs	8-9
Figure 8-8	Cross-stream kerning	8-9
Figure 8-9	Partially and fully inhibiting kerning	8-10
Figure 8-10	Tracking with track settings	8-11
Figure 8-11	Advance widths, including side bearings to allow for interglyph spacing	8-11
Figure 8-12	Misalignment caused by advance widths that vary with glyph size	8-12
Figure 8-13	How curved letters extend below the baseline to align with straight letters	8-12
Figure 8-14	Apparent misalignment of curved letters	8-13
Figure 8-15	The optical edges of a glyph	8-13
Figure 8-16	Optical alignment at line edges	8-14
Figure 8-17	Automatic hanging punctuation	8-14
Figure 8-18	Effects of hanging inhibit factor	8-15
Figure 8-19	Defining a nonhanging glyph as a hanging glyph	8-15
Figure 8-20	A style run with an imposed width in a line of text	8-16
Figure 8-21	Application-specified kerning adjustments	8-17
Figure 8-22	Application-controlled glyph substitution	8-18
Table 8-1	Examples of feature types	8-19
Table 8-2	Feature selectors for the <code>allTypographicFeaturesType</code> font feature type	8-22
Figure 8-23	Ligatures in Roman text	8-23
Figure 8-24	A ligature in Arabic text	8-23
Figure 8-25	Versions of the Arabic letter “ha”	8-24
Table 8-3	Feature selectors for the <code>ligaturesType</code> feature type	8-24
Figure 8-26	Levels of ligature formation controlled with ligature feature selectors	8-25
Figure 8-27	Use of diphthong ligatures	8-25
Figure 8-28	Noncontextual cursive connection in a Roman font	8-26
Table 8-4	Feature selectors for the <code>cursiveConnectionType</code> feature type	8-26
Table 8-5	Feature selectors for the <code>letterCaseType</code> feature type	8-26
Figure 8-29	Case conversion	8-27
Table 8-6	Feature selectors for the <code>verticalSubstitutionType</code> feature type	8-27
Figure 8-30	Vertical substitution forms in a font	8-28
Table 8-7	Feature selectors for the <code>linguisticRearrangementType</code> feature type	8-28
Figure 8-31	The word “hindi” drawn with rearrangement turned on (upper) and off (lower)	8-29
Figure 8-32	Specifying different swashes with feature selectors	8-30
Table 8-8	Feature selectors for the <code>smartSwashType</code> feature type	8-30
Figure 8-33	Hebrew text with diacritical marks shown (upper) and hidden (lower)	8-31
Table 8-9	Feature selectors for the <code>diacriticsType</code> feature type	8-31
Figure 8-34	Accented forms	8-32
Table 8-10	Feature selectors for the <code>verticalPositionType</code> feature type	8-32
Figure 8-35	Fractions	8-33
Table 8-11	Feature selectors for the <code>fractionsType</code> feature type	8-33
Figure 8-36	Allowing and preventing glyph overlap	8-34

Table 8-12	Feature selectors for the <code>overlappingCharactersType</code> feature type	8-34
Figure 8-37	Traditional and simplified versions of a Chinese character	8-35
Table 8-13	Feature selectors for the <code>characterShapeType</code> feature type	8-35
Table 8-14	Feature selectors for the <code>numberSpacingType</code> feature type	8-35
Figure 8-38	Fixed-width and proportional-width numerals	8-36
Figure 8-39	Uppercase and lowercase numerals	8-36
Table 8-15	Feature selectors for the <code>numberCaseType</code> feature type	8-36
Table 8-16	Feature selectors for the <code>styleOptionsType</code> feature type	8-37
Table 8-17	Feature selectors for the <code>typographicExtrasType</code> feature type	8-37
Table 8-18	Feature selectors for the <code>mathematicalExtrasType</code> feature type	8-38
Figure 8-40	Ornamental glyphs	8-39
Table 8-19	Feature selectors for the <code>ornamentSetsType</code> feature type	8-39
Table 8-20	Feature selectors for the <code>characterAlternativesType</code> feature type	8-40
Table 8-21	Feature selectors for the <code>designComplexityType</code> feature type	8-40
Listing 8-1	Setting up a style object for a layout shape	8-41
Listing 8-2	A sample that specifies with-stream and cross-stream shifting	8-43
Figure 8-41	Result of with-stream and cross-stream shift applied to a style run	8-44
Listing 8-3	Using track settings to spread or compress text	8-45
Listing 8-4	Preventing optical alignment	8-46
Listing 8-5	Inhibiting hanging punctuation	8-47
Listing 8-6	Creating a line containing a style run with an imposed width	8-48
Listing 8-7	Adjusting the kerning amount for a pair of glyphs	8-50
Listing 8-8	Using glyph substitutions to replace one glyph with another	8-52
Listing 8-9	Specifying three levels of ligature formation	8-53
Listing 8-10	Specifying three different types of swashes	8-55
Listing 8-11	Specifying three different kinds of case substitution	8-56

Chapter 9

Layout Line Control 9-1

Figure 9-1	Baseline positions for two fonts	9-6
Figure 9-2	How the same glyphs can align to different baselines	9-7
Figure 9-3	Text with multiple baselines aligned to $y = 0$	9-7
Figure 9-4	Preferred alignment for multiple baselines	9-8
Figure 9-5	Creating vertical text in a layout shape	9-9
Figure 9-6	Rotating vertical text in a layout shape	9-9
Figure 9-7	Line length and line span	9-10
Figure 9-8	Factors in line breaking	9-12
Figure 9-9	How glyph direction affects display order	9-14
Figure 9-10	How dominant direction affects display order	9-16

Figure 9-11	The levels array property of the layout shape	9-17
Figure 9-12	How nesting level relates to text direction	9-18
Figure 9-13	Multiple nesting direction levels in one line	9-19
Figure 9-14	Justification gap	9-21
Table 9-1	Justification priorities	9-22
Figure 9-15	Justification-related properties of the style object	9-25
Listing 9-1	Aligning baselines to create drop capitals	9-28
Figure 9-16	Drop capitals created by aligning baselines	9-30
Listing 9-2	Creating and drawing vertical text	9-31
Figure 9-17	Rotated Roman glyphs in vertical text	9-32
Listing 9-3	Breaking a Roman layout shape into individual lines of a paragraph	9-34
Listing 9-4	Defining nested direction levels for a line of text	9-39
Figure 9-18	A text line with nested direction levels	9-41
Listing 9-5	Overriding the glyph direction in a style run	9-42
Figure 9-19	Results of overriding glyph direction	9-43
Listing 9-6	A simple justification example	9-43
Figure 9-20	Unjustified (upper) and justified (lower) lines of different lengths	9-45
Listing 9-7	Displaying partial justification using white space	9-46
Figure 9-21	Five degrees of justification with white space	9-48
Listing 9-8	Displaying partial justification with kashidas	9-48
Figure 9-22	Five degrees of justification with kashidas	9-49
Figure 9-23	Glyph stretching during increasing justification	9-50
Figure 9-24	Ligature decomposition during increasing justification	9-51
Listing 9-9	Overriding justification priorities	9-52
Figure 9-25	Results of justification priority overrides on intercharacter and interword spacing	9-55
Listing 9-10	Overriding justification behavior of the whitespace glyph	9-56
Figure 9-26	Results of overriding justification behavior of the whitespace glyph	9-57

Chapter 10

Layout Carets, Highlighting, and Hit-Testing 10-1

Figure 10-1	Positioning conventions for source text and display text	10-4
Figure 10-2	Edge offsets and glyph indexes in mixed-direction text	10-5
Figure 10-3	Insertion point and caret	10-7
Figure 10-4	Angled and straight carets in single-direction text	10-8
Figure 10-5	Split caret and single carets at a direction boundary in mixed-direction text	10-9
Figure 10-6	Split caret with linguistically rearranged glyphs	10-10
Figure 10-7	Caret positions with and without ligature splits	10-11
Figure 10-8	Moving the caret with Left and Right Arrow keys	10-12
Figure 10-9	Highlighting in single-direction text	10-13
Figure 10-10	Discontiguous visual highlighting in mixed-direction text	10-14
Figure 10-11	Contiguous visual highlighting in mixed-direction text	10-15
Figure 10-12	Hit point and caret position in hit-testing	10-16
Figure 10-13	Hit-testing in mixed-direction text	10-17
Figure 10-14	Projecting the hit point to the baseline	10-18
Listing 10-1	Drawing angled and straight carets at all caret positions	10-19

Figure 10-15	Drawing all possible caret positions in a layout shape's text	10-20
Listing 10-2	Drawing three different types of caret at one edge offset	10-21
Figure 10-16	Drawing different caret types at a single edge offset	10-22
Listing 10-3	A key-down handler using <code>GXGetRightVisualOffset</code> and <code>GXGetLeftVisualOffset</code>	10-23
Listing 10-4	Preventing ligature splits for caret positioning	10-24
Figure 10-17	All caret positions drawn with (upper) and without (lower) ligature splits	10-25
Figure 10-18	Contiguous highlighting from offsets 4 to 13 in single-direction text	10-26
Figure 10-19	Discontiguous highlighting from offsets 4 to 19 in mixed-direction text	10-27
Figure 10-20	Contiguous highlighting from offsets 4 to 19 in mixed-direction text	10-28
Figure 10-21	<code>GXHitTestLayout</code> example	10-29
Listing 10-5	Using the <code>GXHitTestLayout</code> function	10-31
Figure 10-22	Caret positions and glyph indexes for one display version of the word "office"	10-33
Listing 10-6	Converting an edge offset to a glyph index	10-34
Figure 10-23	Using <code>GXGetOffsetGlyphs</code> to locate glyphs corresponding to known offsets	10-37
Listing 10-7	Converting a glyph index to an edge offset	10-38
Figure 10-24	Using <code>GXGetGlyphOffset</code> to locate a glyph's character	10-40