

Figures, Tables, and Listings

Color Plates

Color plates are immediately preceding the title page

Color Plate 1	Writing systems of the world
Color Plate 2	Text control panel
Color Plate 3	Keyboard menu
Color Plate 4	Keyboard icons

Chapter 1

Introduction to Text on the Macintosh 1-1

Figure 1-1	Separation of input, storage, and display in Macintosh text handling	1-4
Figure 1-2	How QuickDraw draws text	1-6
Figure 1-3	Bytes, character codes, characters, and glyphs	1-8
Figure 1-4	Four bytes displayed in Japanese and in English	1-9
Figure 1-5	Storage order and display order	1-10
Figure 1-6	Style runs in text	1-11
Figure 1-7	Key translation (simplified)	1-12
Figure 1-8	Key Caps display of Thai keyboard layout (no modifier keys pressed)	1-13
Figure 1-9	Key Caps display of Cyrillic keyboard layout (Caps Lock key pressed)	1-13
Figure 1-10	TextEdit edits and displays mixed-directional text in a dialog box	1-17
Figure 1-11	Writing-system examples	1-22
Figure 1-12	Words with alphabetic, syllabic, and ideographic characters	1-22
Figure 1-13	Thai character cluster	1-23
Figure 1-14	Line directions in text	1-24
Figure 1-15	Text alignment	1-25
Figure 1-16	Justification through interword (Hebrew) and intercharacter (Japanese) spacing	1-25
Figure 1-17	Justification with Arabic extension bar characters	1-26
Figure 1-18	Contextual forms in cursive English	1-26
Figure 1-19	Standalone and contextual forms in Arabic	1-27
Figure 1-20	A ligature in Roman text	1-27
Figure 1-21	A ligature in Arabic text	1-27
Figure 1-22	A complex ligature in Arabic text	1-28
Figure 1-23	Character reordering in the Devanagari writing system	1-29
Figure 1-24	Arabic text with diacritical mark to specify extra emphasis on a consonant	1-29
Figure 1-25	Vowel marks in Hebrew text	1-29
Figure 1-26	Word demarcation in the Roman writing system	1-30
Figure 1-27	Line breaking in a bidirectional writing system	1-31
Figure 1-28	Word demarcation in Japanese	1-31
Figure 1-29	Selected valid styles in various writing systems	1-32

Figure 1-30	Standard international formats	1-33
Figure 1-31	Components of the script management system for text display	1-36
Table 1-1	The international resources	1-41
Table 1-2	The keyboard resources	1-43
Figure 1-32	Types of script systems	1-47
Figure 1-33	How the script management system handles different types of scripts	1-48
Figure 1-34	The script, language, and region hierarchy	1-49
Figure 1-35	Distinguishing scripts by resource ID range (for script codes 0–32)	1-50
Figure 1-36	The Standard Roman character set	1-55
Figure 1-37	Character encodings for 1-byte script systems	1-57
Figure 1-38	Character encoding for a 2-byte script system (Japanese)	1-59
Figure 1-39	Constructing blocks (Hangul) from elements (Jamo) in Korean	1-60
Figure 1-40	Storage order and display order	1-66
Figure 1-41	How primary line direction affects display order	1-67
Figure 1-42	Dialog items truncated at dialog-box boundary	1-69
Figure 1-43	Style runs, font runs, script runs, and direction runs in text	1-71
Figure 1-44	Caret position and insertion point	1-75
Figure 1-45	Caret positions at direction boundaries	1-76
Figure 1-46	Dual caret at direction boundaries in mixed-directional text	1-78
Figure 1-47	Single carets at direction boundaries in mixed-directional text	1-79
Figure 1-48	Highlighting a selection range in unidirectional text	1-80
Figure 1-49	Highlighting a selection range in mixed-directional text	1-81
Figure 1-50	Interpreting caret position from a mouse-down event	1-83
Figure 1-51	Mouse-down regions and caret positions in mixed-directional text	1-84
Figure 1-52	Apple Keyboard II (domestic layout)	1-87
Figure 1-53	Key translation	1-89
Figure 1-54	Font script and keyboard script synchronization	1-90
Figure 1-55	Bottomline input window for Japanese input method	1-92
Figure 1-56	Active input area (underlined) for inline input	1-92
Figure 1-57	Bottomline input in Korean	1-93
Figure 1-58	Filenames and dates in Arabic and U.S. formats (Arabic system script)	1-96
Figure 1-59	System-script components in the System Folder	1-102
Figure 1-60	Menu bar with keyboard icon	1-105
Figure 1-61	Keyboard icons and input-method icons	1-105
Figure 1-62	Keyboard menu	1-106
Figure 1-63	Arabic Key Caps	1-107
Figure 1-64	Text control panel	1-108
Figure 1-65	Numbers control panel	1-109
Figure 1-66	Date & Time control panel	1-110
Figure 1-67	Date Formats dialog box (from Date & Time control panel)	1-110
Figure 1-68	Time Formats dialog box (from Date & Time control panel)	1-111

Figure 2-1	Style runs in a line of text	2-8
Figure 2-2	Mixed-directional text display	2-8
Figure 2-3	Discontinuous highlighting display	2-10
Figure 2-4	Outline highlighted text selection in background window	2-11
Figure 2-5	Caret movement across a direction boundary	2-12
Figure 2-6	Destination and view rectangles	2-16
Figure 2-7	Relationship between the TextEdit data structures for monostyled text	2-19
Figure 2-8	Relationships among the TextEdit data structures for multistyled text	2-20
Listing 2-1	Using <code>TETextBox</code> to draw static text	2-25
Listing 2-2	A sample document record	2-26
Listing 2-3	Creating a multistyled edit record	2-29
Listing 2-4	An idle-processing procedure	2-34
Listing 2-5	Passing a mouse-down event to TextEdit	2-36
Listing 2-6	Inserting text in a document	2-38
Listing 2-7	Getting the selection range length	2-39
Figure 2-9	Cutting text from a multistyled edit record	2-41
Listing 2-8	Handling Cut, Copy, and Paste commands on an Edit menu	2-42
Figure 2-10	Continuous attributes over a selection range	2-45
Listing 2-9	Determining the font, style, size, and color of the current selection range	2-46
Figure 2-11	An initial selection before <code>TESetStyle</code> is called	2-47
Figure 2-12	The result of calling <code>TESetStyle</code> to toggle to bold	2-47
Figure 2-13	The result of calling <code>TESetStyle</code> to toggle italics	2-48
Listing 2-10	Handling the Font menu	2-48
Listing 2-11	Handling the Size menu	2-49
Listing 2-12	Handling a Style menu	2-49
Listing 2-13	Checking the style and marking Style menu items to reflect the current selection range	2-51
Listing 2-14	Saving a multistyled text edit record to disk	2-53
Listing 2-15	Restoring a document that uses multistyled TextEdit	2-55
Listing 2-16	Checking for 2-byte characters when backspacing	2-57
Figure 2-14	Determining when to use <code>WIDTHHook</code> and <code>nWIDTHHook</code>	2-60
Figure 2-15	The TextEdit data structures and fields	2-67

Figure 3-1	Stylistic variations	3-7
Figure 3-2	Effect of the basic transfer modes for black-and-white images	3-9
Figure 3-3	Multiple style runs on a single line	3-11
Figure 3-4	Justification of Roman text	3-14
Listing 3-1	Using QuickDraw to set the graphics port text-related fields	3-20
Listing 3-2	Calling <code>StyledLineBreak</code> to identify where to break the text line	3-31
Listing 3-3	An application-defined run direction function called by <code>GetFormatOrder</code>	3-35

Listing 3-4	Determining the style run display order and drawing the line	3-36
Figure 3-5	Calling <code>VisibleLength</code> for a Roman style run	3-37
Figure 3-6	Calling <code>VisibleLength</code> for a Hebrew style run	3-38
Figure 3-7	Calling <code>VisibleLength</code> for Hebrew text with Roman space characters	3-38
Listing 3-5	Distributing slop value among style runs	3-41
Listing 3-6	Calling <code>GetFontInfo</code> to determine the line height	3-43
Listing 3-7	Turning off reordering of right-to-left text before calling <code>PixelToChar</code> for line-breaking	3-45
Listing 3-8	Using <code>StdTxMeas</code> to get the font metrics for determining the line height of scaled text	3-46
Figure 3-8	What pixel position means for <code>CharToPixel</code> and <code>PixelToChar</code>	3-48
Figure 3-9	Caret position for a leading-edge mouse-down event	3-50
Figure 3-10	Caret position for a trailing-edge mouse-down event	3-51
Figure 3-11	Caret position for a leading-edge mouse-down event at a direction boundary	3-52
Figure 3-12	Caret position for a trailing-edge mouse-down event at a direction boundary	3-53
Figure 3-13	Caret position for a trailing-edge mouse-down event at a direction boundary	3-54
Figure 3-14	Caret position for a mouse-down event beyond the last glyph of the text segment	3-55
Listing 3-9	Drawing the caret and highlighting a selection range	3-55
Figure 3-15	Highlighting mixed-directional text	3-60
Listing 3-10	Generating a picture file with font information	3-64
Listing 3-11	A picture file with font information	3-64
Table 3-1	Effects of the basic transfer modes	3-71
Table 3-2	Transfer mode constants and selectors	3-73

Chapter 4

Font Manager		4-1
Figure 4-1	Terms for font measurements	4-9
Figure 4-2	The ascent line and maximum y-value	4-11
Figure 4-3	Unkerned text (top) and kerned text (bottom)	4-12
Table 4-1	Subdivisions of Roman font family IDs	4-14
Figure 4-4	A comparison of scaled bitmapped and outline fonts	4-19
Figure 4-5	A glyph stretched horizontally	4-20
Figure 4-6	A glyph stretched vertically	4-21
Figure 4-7	A glyph condensed horizontally	4-21
Figure 4-8	The effect of an off-curve point on two Bézier curves	4-26
Figure 4-9	An outline with points on and off the curve	4-27
Figure 4-10	A curve with consecutive off-curve points	4-28
Figure 4-11	A glyph from an outline font	4-29
Figure 4-12	An unmodified glyph from an outline font at a small point size	4-30
Figure 4-13	An instructed glyph from an outline font	4-31
Figure 4-14	A sample Size menu and font size dialog box	4-32
Listing 4-1	Checking a font name against the system font name	4-33

Figure 4-15	The difference between a scaled glyph and a preserved glyph	4-36
Figure 4-16	The bitmapped font ('NFNT') resource	4-67
Figure 4-17	The font directory	4-74
Listing 4-2	Calculating the checksum of a given table	4-76
Figure 4-18	A glyph description	4-78
Figure 4-19	The font header table	4-79
Listing 4-3	Calculating the checksum of a font	4-81
Figure 4-20	The horizontal metrics table	4-83
Figure 4-21	The naming table	4-85
Table 4-2	Platform identifiers	4-86
Table 4-3	ISO platform-specific identifiers	4-87
Table 4-4	ISO language codes	4-87
Table 4-5	Font name identifiers	4-88
Figure 4-22	The font family ('FOND') resource	4-91
Figure 4-23	Style codes	4-95
Figure 4-24	The font association table	4-95
Figure 4-25	The offset table	4-96
Figure 4-26	The bounding-box table	4-97
Figure 4-27	The font family glyph-width table	4-98
Figure 4-28	The style-mapping table	4-100
Listing 4-4	Using the style-mapping table to build a PostScript font name	4-103
Figure 4-29	The font family kerning table	4-106
Figure 4-30	A kerning pair entry	4-107

Chapter 5

Text Utilities 5-1

Figure 5-1	Determining the current script	5-5
Figure 5-2	A string containing 1-byte and 2-byte characters	5-7
Listing 5-1	Using the <code>NewString</code> and <code>SetString</code> routines	5-8
Figure 5-3	Strings in different languages in one list	5-11
Figure 5-4	Strings in different languages sorted by script	5-11
Figure 5-5	Strings in different languages sorted by language within script	5-12
Table 5-1	Excerpt from the Standard Roman script system sorting order	5-13
Figure 5-6	Choosing a string comparison routine	5-16
Table 5-2	Sorting features of the Macintosh file system	5-17
Figure 5-7	Truncating a pathname in its middle	5-20
Listing 5-2	Truncating a pathname	5-20
Figure 5-8	Replacing a portion of a string with 1-byte and 2-byte characters	5-21
Listing 5-3	Substituting and truncating text	5-22
Figure 5-9	Finding line breaks in multiscript text	5-25
Figure 5-10	Relationships of the parameters of <code>StyledLineBreak</code>	5-26
Listing 5-4	Using the <code>StyledLineBreak</code> function	5-27
Figure 5-11	Extracting blocks of Roman text	5-28
Table 5-3	Variations in time and short date formats	5-29
Table 5-4	Variations in long and abbreviated date formats	5-30
Listing 5-5	Using <code>StringToDate</code> and <code>StringToTime</code>	5-31

Table 5-5	<code>StringToDateStatus</code> values and their meanings	5-33
Figure 5-12	Using the number formatting routines	5-37
Table 5-6	<code>FormatResultType</code> values for numeric conversion functions	5-38
Listing 5-6	Converting a long integer into a numeric string	5-39
Table 5-7	Numeric string formats	5-39
Table 5-8	Examples of number format specification strings	5-40
Table 5-9	Literals in number format strings	5-41
Table 5-10	Filling digits in	5-42
Table 5-11	Quoting mechanisms in number format strings	5-42
Table 5-12	Symbols in number format strings	5-43
Table 5-13	Implicit language codes	5-55

Chapter 6

Script Manager 6-1

Table 6-1	Evolution of the Script Manager	6-6
Table 6-2	Version numbers for the Script Manager and Roman script system	6-9
Table 6-3	Script Manager variables accessed through <code>GetScriptManagerVariable/</code> <code>SetScriptManagerVariable</code>	6-11
Listing 6-1	Specifying a dual caret with <code>SetScriptManagerVariable</code>	6-13
Table 6-4	Script variables accessed through <code>GetScriptVariable/SetScriptVariable</code>	6-14
Listing 6-2	Representing font names correctly in the script for that font	6-16
Listing 6-3	Setting the size of the Balloon Help font	6-16
Table 6-5	Constants for the <code>code</code> parameter in the <code>KeyScript</code> procedure	6-18
Listing 6-4	Setting the keyboard script from the font script	6-20
Listing 6-5	Setting the font (script) from the keyboard script	6-21
Figure 6-1	Determining script code from font family ID	6-23
Listing 6-6	Handling 2-byte characters in a search procedure	6-27
Figure 6-2	Fields in the <code>CharacterType</code> return value	6-28
Listing 6-7	Determining the number separators for the current script	6-33
Listing 6-8	Getting number parts from a script system's number parts table	6-34
Listing 6-9	Getting a token string from the untoken table	6-36
Figure 6-3	The action of <code>IntlTokenize</code>	6-39
Figure 6-4	<code>IntlTokenize</code> data structures (simplified)	6-40
Figure 6-5	The effects of transliteration	6-45
Figure 6-6	Dispatch table entry for script utilities and <code>QuickDraw</code> patches	6-51
Figure 6-7	Style code format	6-73

Chapter 7

Text Services Manager 7-1

Figure 7-1	Bottomline input with a floating input window	7-7
Figure 7-2	Inline input	7-7
Figure 7-3	Displaying conversion options for bottomline input	7-8
Figure 7-4	How a TSM-aware client application uses the Text Services Manager	7-10
Figure 7-5	Entering, converting, and confirming text in an active input area	7-12
Figure 7-6	How a non-TSM-aware application uses the Text Services Manager	7-13
Figure 7-7	Floating window service layer	7-14
Figure 7-8	The format of the <code>componentFlags</code> field of the component description record	7-16
Listing 7-1	Initializing as a TSM-aware application	7-18
Listing 7-2	Creating a new TSM document and associating it with a window	7-19
Listing 7-3	Activating and deactivating a TSM document	7-21
Listing 7-4	Passing events to a text service component	7-22
Listing 7-5	Confirming text in an active input area	7-23
Listing 7-6	Closing a TSM-aware application	7-24
Listing 7-7	A sample handler for the Update Active Input Area Apple event	7-26
Listing 7-8	A sample handler for the Position To Offset Apple event	7-30
Figure 7-9	Drawing a window with conversion options next to the active input area	7-33
Listing 7-9	A sample handler for the Offset To Position Apple event	7-33
Figure 7-10	Input method icons in the Keyboard menu and menu bar	7-40
Listing 7-10	Determining the script and language for a text service component	7-43
Listing 7-11	Constructing and sending an Update Active Input Area Apple event	7-45
Table 7-1	Apple event ID constants	7-66
Table 7-2	Apple event keyword constants	7-66
Table 7-3	Apple event descriptor types	7-67
Table 7-4	Apple event descriptor type constants for the Apple event region class	7-67
Figure 7-11	Updating text in an active input area	7-69

Chapter 8

Dictionary Manager 8-1

Figure 8-1	General format of a dictionary record	8-5
Figure 8-2	Format of data associated with a key	8-6
Figure 8-3	Format of an entry in the data associated with a key	8-7
Figure 8-4	A simple dictionary with no garbage data	8-8
Figure 8-5	Creating garbage data in a dictionary	8-9
Figure 8-6	Deleting garbage data from a dictionary	8-10
Listing 8-1	Creating a dictionary file	8-12
Listing 8-2	Opening and closing a dictionary file	8-13
Listing 8-3	Obtaining information about a dictionary	8-15

Figure 8-7	The requested attributes table	8-16
Table 8-1	Sample data returned by <code>FindRecordInDictionary</code>	8-16
Listing 8-4	Displaying all records in a dictionary by index	8-17
Listing 8-5	Inserting a record into a dictionary	8-19
Table 8-2	Defined attribute types for dictionary entries	8-27

Appendix A

Built-in Script Support A-1

Figure A-1	The Standard Roman character set	A-5
Table A-1	Nonprinting characters in the Standard Roman character set	A-6
Table A-2	Low-ASCII characters to avoid as delimiters	A-7
Table A-3	Printing characters in the Standard Roman character set	A-9
Table A-4	Croatian variations from the Standard Roman character set	A-16
Table A-5	Romanian variations from the Standard Roman character set	A-17
Table A-6	Turkish variations from the Standard Roman character set	A-18
Table A-7	Icelandic and Faroese variations from the Standard Roman character set	A-18
Table A-8	Standard sorting order (for Standard Roman character set)	A-20
Table A-9	International resources in U.S. system software	A-23
Table A-10	Keyboard resources in U.S. system software	A-24
Table A-11	Script utilities supported by WorldScript I	A-26
Table A-12	QuickDraw patches supported by WorldScript I	A-27
Figure A-2	Dispatch table entry for script utilities and QuickDraw patches	A-29
Figure A-3	How calls are dispatched to the 1-byte script utilities	A-30
Figure A-4	How calls are dispatched to the 1-byte QuickDraw patches	A-31
Table A-13	Classification of 1-byte script utilities by function	A-33
Table A-14	Classification of 1-byte QuickDraw patches by function	A-35
Table A-15	Script utilities supported by WorldScript II	A-37

Appendix B

International Resources B-1

Table B-1	The international resources	B-4
Table B-2	Resource ID ranges for each script system	B-6
Figure B-1	Format of the script-sorting resource header	B-13
Figure B-2	Script, language, and region data tables in the script-sorting resource	B-14
Table B-3	Sorted scripts, languages, and regions from a script-sorting resource	B-15
Table B-4	Constants for specifying numeric separators	B-23
Figure B-3	Examples of long date formatting	B-31
Table B-5	Separator positions in long date format	B-31
Figure B-4	Format of the string-manipulation resource header	B-36
Figure B-5	Format of the script run table header (new format)	B-41
Figure B-6	Script run table state table	B-42
Figure B-7	Format of a script run table action code	B-43
Figure B-8	Format of the script run table return table	B-43

Figure B-9	NBreakTable state table	B-47
Figure B-10	Format of an NBreakTable action code	B-47
Table B-6	Example of classes for an NBreakTable state table	B-48
Table B-7	Example of states for an NBreakTable state table	B-48
Figure B-11	Forward operation of the state machine for word selection	B-50
Figure B-12	Format of the token table	B-53
Figure B-13	Format of the whitespace table	B-58
Figure B-14	Format of the script configuration table	B-61
Table B-8	A script configuration table for a Hebrew encoding/rendering resource	B-62
Figure B-15	Format of the character expansion table	B-64
Figure B-16	Format of the glyph-to-character table	B-65
Figure B-17	Format of the break-table directory	B-66
Table B-9	Sample encoding/rendering resource for a 2-byte script system	B-68
Figure B-18	Format of the transliteration resource header	B-71
Figure B-19	Format of a transliteration rule	B-72

Appendix C

Keyboard Resources C-1

Table C-1	The keyboard types	C-4
Figure C-1	Apple Keyboard II (domestic layout)	C-5
Figure C-2	Apple Extended Keyboard II (domestic layout)	C-5
Table C-2	The keyboard modifier bits in an event record	C-6
Table C-3	The keyboard resources	C-7
Figure C-3	The key translation process	C-10
Figure C-4	Format of the key-map resource	C-12
Table C-4	Key-map resource assignment of raw key codes to virtual key codes	C-13
Table C-5	Reassigning right key codes for Shift, Option, and Control keys	C-14
Table C-6	ADB and non-ADB virtual key codes for cursor keys and keypad keys	C-15
Table C-7	Virtual key codes for the international Macintosh Plus keyboard	C-16
Figure C-5	Format of an entry in the key-remap resource	C-17
Figure C-6	Format of the keyboard-layout resource	C-19
Figure C-7	Inside the keyboard-layout resource	C-21
Listing C-1	Loading a non-system keyboard-layout resource	C-22
Listing C-2	Regenerating a character code with KeyTranslate	C-24
Table C-8	Keyboard color icon types and standard icon equivalents	C-25
Figure C-8	Sample keyboard icons	C-26
Figure C-9	Format of entries in the keyboard-swap resource	C-26
Listing C-3	A hypothetical keyboard-swap resource	C-27
Figure C-10	Format of the key-caps resource	C-29
Listing C-4	Sample key-caps resource data in Rez format	C-31
Figure C-11	Shape array and resulting region for the Return key	C-32
Figure C-12	Key Caps display with key origins	C-33
Figure C-13	Key Caps display of dead keys with Option key pressed	C-33
Figure C-14	Key Caps display of completer keys after circumflex dead key has been pressed	C-34

Table D-1	Renamed, relocated, and obsolete text and international routines	D-4
------------------	--	-----