

# Figures, Tables, and Listings

## Color Plates

---

*Color plates are immediately preceding the title page*

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

## Chapter 1

## Introduction to Text on the Macintosh 1-1

---

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

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

## Chapter 2

### TextEdit 2-1

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

## Chapter 3

### QuickDraw Text 3-1

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

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

## Chapter 4

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

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

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

## Chapter 6

## Script Manager 6-1

---

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

## Chapter 7

### Text Services Manager 7-1

---

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

## Chapter 8

### Dictionary Manager 8-1

---

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

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

## Appendix A

### Built-in Script Support A-1

---

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

## Appendix B

### International Resources B-1

---

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



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

## Appendix C

## Keyboard Resources C-1

---

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

<b>Table D-1</b>	Renamed, relocated, and obsolete text and international routines D-4
------------------	--