

# Contents

Figures, Tables, and Listings    xxi

## Preface

## About This Book    xxxi

---

What to Read    xxxi  
Format of a Typical Chapter    xxxii  
Conventions Used in This Book    xxxiii  
    Special Fonts    xxxiii  
    Types of Notes    xxxiii  
Development Environment    xxxiv  
Developer Products and Support    xxxiv

## Chapter 1

## Introduction to Text on the Macintosh    1-1

---

Macintosh Text Overview    1-3  
    Separation of Tasks    1-4  
    Text Is Graphics    1-5  
    Characters, Glyphs, Character Codes, and Bytes    1-8  
    Text Storage    1-9  
    Keyboards and Input Methods    1-11  
    Writing Systems and Script Systems    1-14  
    Macintosh Text Utilities    1-16  
    TextEdit, a Text-Processing Service    1-16  
    Planning Your Text Handling Capabilities    1-18  
        Rudimentary Text Handling    1-18  
        Moderate Text Handling    1-19  
        Sophisticated Text Handling    1-20  
    Writing Systems and Script Systems    1-21  
    Features of the World's Writing Systems    1-21  
        Character Representation    1-22  
        Line Direction and Alignment    1-23  
        Contextual Forms and Character Reordering    1-26  
        Diacritical Marks    1-29  
        Uppercase and Lowercase Characters    1-30  
        Word Demarcation    1-30  
        Styles    1-31  
        Numbers, Currencies, and Dates    1-32  
        Character Order and Text Sorting    1-34  
        Variations Among Languages and Regions    1-34

Components of the Macintosh Script Management System	1-35
The Macintosh Text Managers	1-36
The WorldScript Extensions	1-39
Components of a Script System	1-40
International Resources	1-41
Keyboard Resources	1-42
Fonts	1-44
How Script Systems Are Classified	1-45
Types of Script Systems	1-46
Script Codes, Language Codes, and Region Codes	1-48
The System Script and Auxiliary Scripts	1-51
Font Script and Keyboard Script	1-51
How Script Systems Work	1-52
Character Encoding	1-52
The Standard Roman Character Set	1-54
Other 1-Byte Character Encodings	1-56
2-Byte Character Encodings	1-57
Font Handling	1-60
Font Availability and Selection	1-61
System Font and Application Font	1-61
Roman Characters and Associated Fonts	1-62
Other Font Issues	1-63
Character Rendering and Text Display	1-64
Storage Order and Display Order	1-65
Line Direction and Alignment	1-67
Style Runs, Font Runs, Script Runs, Direction Runs	1-70
Text Layout	1-71
Caret Handling	1-74
Highlighting	1-80
Converting Screen Position to Text Offset	1-82
Printing	1-85
Text Input	1-87
Keyboards and Key Translation	1-87
Input Methods	1-91
Text Manipulation	1-94
Sorting Strings	1-94
Formatting Dates, Times, Numbers, and Symbols	1-96
Analyzing Characters	1-98
Searching, Modifying, and Converting Text	1-98
Finding Word Boundaries and Line Breaks	1-99
Script Systems in Use	1-100
Installing and Enabling Script Systems	1-100
Components of the System Script	1-101
Components of Auxiliary Scripts	1-102
Installing Modifications to a Script System	1-103
How the User Switches Among Script Systems	1-104
User Control of Script Settings	1-107

About TextEdit	2-6
TextEdit and Standard Macintosh Features	2-6
Multistyled and Monostyled Text	2-7
Font and Keyboard Script Synchronization	2-8
Cutting, Copying, and Pasting Text	2-9
The TextEdit User Interface	2-10
The Selection Range, the Insertion Point, and Highlighting in TextEdit	2-10
Caret Position and Movement	2-11
Text Alignment	2-13
Line Measurement	2-14
Text Buffering	2-14
The TextEdit Private, Null, and Style Scraps	2-15
An Overview of the TextEdit Data Structures	2-16
An Overview of the Edit Record	2-16
Related Data Structures	2-17
Using TextEdit	2-22
Getting Started With TextEdit	2-23
Preparing to Use TextEdit	2-23
Displaying Static Text	2-25
Creating an Edit Record	2-25
Specifying the Destination and View Rectangles	2-29
Setting the Text of an Edit Record	2-30
Setting the Selection Range or the Insertion Point	2-31
Scrolling Text	2-32
Disposing of an Edit Record	2-33
Responding to Events Using TextEdit	2-33
Handling a Null Event	2-33
Activating an Edit Record	2-34
Handling Mouse-Down Events	2-35
Responding to an Update Event	2-37
Accepting Text Input Through Key-Down Events	2-37
Moving Text In and Out of Edit Records	2-39
Using TextEdit to Cut, Copy, and Paste Text	2-40
Inserting and Deleting Text	2-44
Text Attributes	2-44
Checking the Text Attributes Across a Selection Range	2-45
Toggling an Attribute	2-47
Handling a Font Menu	2-48
Handling a Font Size Menu	2-49
Handling a Style Menu	2-49
Changing the Text Alignment	2-51
Saving and Restoring a TextEdit Document, and Implementing Undo	2-52
Saving a TextEdit Document	2-52

Restoring an Existing TextEdit Document	2-54
Handling Undo	2-56
Customizing TextEdit	2-57
Replacing the End-of-Line Routine	2-58
Replacing the Drawing Routine	2-59
Replacing the Width-Measuring Routines	2-59
Replacing the Hit Test Routine	2-61
Customizing Word Selection	2-61
Customizing Automatic Scrolling	2-62
Determining the Line Length	2-63
Advanced Customization	2-64
TextEdit Reference	2-65
Data Structures	2-65
The Edit Record	2-69
The High Hook and Caret Hook Fields	2-72
The Style Record	2-73
The Style Table	2-74
The Line Height Table	2-75
The Null Style Record	2-76
The Style Scrap Record	2-76
The Scrap Style Table	2-77
Text Style Record	2-78
Routines	2-78
Initializing TextEdit, Creating an Edit Record, and Disposing of an Edit Record	2-79
Activating and Deactivating an Edit Record	2-82
Setting and Getting an Edit Record's Text and Character Attribute Information	2-83
Setting the Caret and Selection Range	2-86
Displaying and Scrolling Text	2-88
Modifying the Text of an Edit Record	2-95
Managing the TextEdit Private Scrap	2-100
Checking, Setting, and Replacing Styles	2-101
Using Byte Offsets and Corresponding Points	2-107
Additional TextEdit Features	2-109
Customizing TextEdit	2-112
Summary of TextEdit	2-120
Pascal Summary	2-120
Constants	2-120
Data Types	2-122
Routines	2-125
C Summary	2-127
Constants	2-127
Types	2-129
Routines	2-132

Assembly-Language Summary	2-134
Trap Macros	2-134
Global Variables	2-135

## Chapter 3

## QuickDraw Text 3-1

---

About QuickDraw Text	3-4
Graphics Ports and Text Drawing	3-4
Font, Font Style, and Font Size	3-5
Transfer Modes	3-8
QuickDraw Text, Script Systems, and Other Managers	3-10
Text Formatting and Justification	3-13
Scaling	3-15
Caret and Highlighting	3-16
Using QuickDraw Text	3-17
Preparing to Use QuickDraw	3-17
Determining the Version and Initializing QuickDraw	3-18
Setting Up the Text-Drawing Environment	3-19
Specifying Text Characteristics	3-19
Setting the Font	3-20
Modifying the Text Style	3-21
Changing the Font Size	3-22
Changing the Width of Characters	3-22
Using Fractional Glyph Widths	3-23
Specifying the Transfer Mode	3-24
Basic Transfer Mode Operations	3-24
Arithmetic Transfer Mode Operations	3-25
The grayishTextOr Transfer Mode	3-26
Text Mask Mode	3-26
Transparent Transfer Mode	3-27
Transfer Modes and Multibit Fonts	3-27
Measuring and Drawing Single Segments of Text	3-27
Individual Glyphs	3-28
Pascal Strings	3-28
Text Segments	3-29
Measuring and Drawing Lines of Text	3-29
Determining Where to Break the Line	3-30
Determining the Display Order for Style Runs	3-33
Eliminating Trailing Spaces (for Justified Text)	3-36
Calculating the Slop Value (for Justified Text)	3-39
Allocating the Slop to Each Style Run (for Justified Text)	3-39
Drawing the Line of Text	3-42
Using Scaled Text	3-44
Drawing Carets and Highlighting	3-47
Converting an Onscreen Pixel Location to a Byte Offset	3-49
Finding a Caret Position and Drawing a Caret	3-49

Synchronizing the Caret With the Keyboard Script	3-59
Highlighting a Text Selection	3-60
Customizing QuickDraw's Text Handling	3-62
Text in QuickDraw Pictures	3-63
Fonts	3-63
Text With Multiple Style Runs	3-65
QuickDraw Text Reference	3-65
Data Structures	3-66
The Font Information Record	3-66
The Style Data Type	3-66
Routines	3-67
Setting Text Characteristics	3-68
Drawing Text	3-76
Measuring Text	3-81
Laying Out a Line of Text	3-87
Determining the Caret Position, and Selecting and Highlighting Text	3-91
Low-Level QuickDraw Text Routines	3-98
Application-Supplied Routine	3-100
Summary of QuickDraw Text	3-102
Pascal Summary	3-102
Constants	3-102
Data Types	3-102
Routines	3-103
C Summary	3-105
Constants	3-105
Types	3-105
Routines	3-106
Assembly-Language Summary	3-107
Trap Macros	3-107
Global Variables	3-108

## Chapter 4

## Font Manager 4-1

---

About Fonts	4-6
Characters, Character Codes, and Glyphs	4-6
Kinds of Fonts	4-7
Identifying Fonts	4-8
Font Measurements	4-8
About Font Resources	4-12
Font Resource Types	4-13
A Brief History of Font Resource Use	4-13
Font Family IDs	4-14
Restrictions on the Use of 'FONT' Resources	4-15
Font Resource Tables	4-16

About the Font Manager	4-16
How QuickDraw Requests a Font	4-16
How the Font Manager Responds to a Font Request	4-17
How the Font Manager Scales Fonts	4-19
The Scaling Process for a Bitmapped Font	4-22
The Scaling Process for an Outline Font	4-23
How the Font Manager Calculates Glyph Widths	4-23
Synthetic Fonts	4-25
How the Font Manager Renders Outline Fonts	4-25
Using the Font Manager	4-31
Adding Font Sizes and Names to the Menu	4-32
Storing a Font Name in a Document	4-33
Getting Font Measurement Information	4-34
Favoring Outline or Bitmapped Fonts	4-35
Preserving the Shapes of Glyphs	4-35
Using Width Tables	4-36
Getting the System or Application Font ID	4-38
Using Fractional Glyph Widths and Font Scaling	4-38
Font Manager Reference	4-39
Data Structures	4-39
The Font Input Record	4-40
The Font Output Record	4-41
The Global Width Table	4-43
The Font Record	4-46
The Font Family Record	4-47
The Font Association Table Record	4-47
The Family Glyph-Width Table Record	4-48
The Style-Mapping Table Record	4-49
The Font Family Kerning Table Record	4-49
Routines	4-50
Initializing the Font Manager	4-50
Getting Font Information	4-51
Using the Current, System, and Application Fonts	4-53
Getting the Characteristics of a Font	4-54
Enabling Fractional Glyph Widths	4-58
Disabling Font Scaling	4-59
Favoring Outline Fonts Over Bitmapped Fonts	4-60
Scaling Outline Fonts	4-62
Accessing Information About a Font	4-64
Handling Fonts in Memory	4-65
The Bitmapped Font ('NFNT') Resource	4-66
The Font Type Element	4-70
The Offset to the Width/Offset Table	4-71
The Outline Font ('sfnt') Resource	4-72
The Font Directory	4-74
The Character-Code Mapping Table	4-76

The Control-Value Table	4-77
The Font Program Table	4-77
The Glyph Data Table	4-77
The Horizontal Device Metrics Table	4-78
The Font Header Table	4-79
The Horizontal Header Table	4-83
The Horizontal Metrics Table	4-83
The Kerning Table	4-84
The Location Table	4-84
The Maximum Profile Table	4-84
The Font Naming Table	4-85
The PostScript Table	4-89
The Preprogram Table	4-89
The Font Family ('FOND') Resource	4-90
The Font Style Code	4-94
The Font Association Table	4-95
The Offset Table	4-96
The Bounding-Box Table	4-97
The Family Glyph-Width Table	4-98
The Style-Mapping Table	4-99
The Font Family Kerning Table	4-106
Summary of the Font Manager	4-108
Pascal Summary	4-108
Constants	4-108
Data Types	4-108
Routines	4-112
C Summary	4-113
Constants	4-113
Data Types	4-114
Routines	4-118
Assembly-Language Summary	4-119
Trap Macros	4-119
Global Variables	4-120

## Chapter 5

## Text Utilities 5-1

---

About the Text Utilities	5-3
The Text Utilities and the International Resources	5-4
Obtaining Resource Information	5-4
Pascal Strings and Text Strings	5-6
Using the Text Utilities	5-7
Defining Strings	5-8
Working With String Handles	5-8
Working With String Resources	5-9



Sorting Strings in Different Languages	5-9
Sorting Strings in the Same Language	5-12
Primary and Secondary Sorting Order	5-12
Expansion and Contraction of Characters	5-14
Ignorable Characters	5-14
Converting and Stripping Characters	5-14
Special Cases for Sorting	5-14
Variations in Sorting Behavior	5-15
Choosing a Comparison Routine	5-15
Testing Two Strings for Equality	5-17
Comparing Two Strings for Ordering	5-18
Modifying Text	5-18
Converting Characters and Stripping Marks in Strings	5-19
Fitting a String Into a Screen Area	5-19
Replacing a Portion of a String	5-21
Finding Word, Line, and Script Run Boundaries	5-23
Finding Word Boundaries	5-23
Finding Line Breaks	5-24
Finding Subscripts Within a Script Run	5-28
Working With Date and Time Strings	5-29
Converting Formatted Date and Time Strings	
Into Internal Numeric Representations	5-31
Date and Time Value Representations	5-34
Converting Standard Date and Time Values Into Strings	5-34
Working With Numeric Strings	5-35
Converting Between Integers and Numeric Strings	5-38
Using Number Format Specification Strings	5-39
Converting Number Format Specification Strings Into Internal	
Numeric Representations	5-43
Converting Between Floating-Point Numbers and Numeric	
Strings	5-43
Text Utilities Reference	5-44
Data Structures	5-44
Routines	5-47
Defining and Specifying Strings	5-47
Comparing Strings for Equality	5-50
Determining Sorting Order for Strings in Different Languages	5-54
Determining Sorting Order for Strings in the Same Language	5-59
Modifying Characters and Diacritical Marks	5-64
Truncating Strings	5-71
Searching for and Replacing Strings	5-74
Working With Word, Script, and Line Boundaries	5-77
Converting Date and Time Strings Into Numeric Representations	5-82
Converting Numeric Representations Into Date and Time Strings	5-86
Converting Long Date and Time Values Into Strings	5-89
Converting Between Integers and Strings	5-91

Using Number Format Specification Strings for International Number Formatting	5-94
Converting Between Strings and Floating-Point Numbers	5-98
Summary of Text Utilities	5-102
Pascal Summary	5-102
Constants	5-102
Data Types	5-103
Routines	5-104
C Summary	5-107
Constants	5-107
Types	5-109
Routines	5-110
Assembly-Language Summary	5-113
Trap Macros	5-113

## Chapter 6

## Script Manager 6-1

---

About the Script Manager	6-3
The Script Manager and the Script Management System	6-4
The Script Manager and Applications	6-4
Evolution of the Script Manager	6-6
Using the Script Manager	6-7
Testing for the Script Manager and Script Systems	6-8
Controlling Settings	6-10
Checking and Setting the System Direction	6-10
Checking and Setting Script Manager Variables	6-11
Checking and Setting Script Variables	6-13
Making Keyboard Settings	6-17
Synchronizing the Font Script and Keyboard Script	6-19
Obtaining Information	6-21
Determining Script Codes From Font Information	6-21
Analyzing Characters	6-26
Directly Accessing International Resources	6-31
Using Currency, Number, and Date Formats	6-33
Using Number Parts	6-34
Retrieving Text From Tokens	6-35
Using Word-Break Tables	6-37
Using Whitespace Information	6-37
Converting Text	6-37
Tokenization	6-38
Transliteration	6-43
Modifying Script Systems	6-48
Replacing a Script System's Default International Resources	6-48
Replacing a Script System's Default Routines	6-50

Script Manager Reference	6-52
Constants	6-52
Script Codes	6-52
Language Codes	6-54
Region Codes	6-57
Token Codes	6-58
Selectors for Script Manager Variables	6-61
Selectors for Script Variables	6-65
Data Structures	6-73
Token Block Record	6-74
Token Record	6-74
Routines	6-75
Checking and Setting the System Direction	6-76
Checking and Setting Script Manager Variables	6-77
Checking and Setting Script Variables	6-78
Making Keyboard Settings	6-80
Determining Script Codes From Font Information	6-81
Analyzing Characters	6-84
Directly Accessing International Resources	6-89
Tokenization	6-92
Transliteration	6-98
Replacing a Script System's Default Routines	6-101
Summary of the Script Manager	6-107
Pascal Summary	6-107
Constants	6-107
Data Types	6-121
Routines	6-122
C Summary	6-124
Constants	6-124
Data Types	6-124
Routines	6-125
Assembly-Language Summary	6-127
Trap Macros	6-127
Global Variables	6-127

---

<b>Chapter 7</b>	<b>Text Services Manager</b>	<b>7-1</b>
------------------	------------------------------	------------

---

About Text Services	7-6
About Input Methods	7-6
About the Text Services Manager	7-9
The Text Services Environment	7-9
The Text Services Manager and Input Methods	7-11
Inline Input	7-11
Floating Input Windows	7-13
Floating Utility Windows	7-14
About Text Service Components	7-14

Using the Text Services Manager (for Client Applications)	7-17
Testing for the Availability of the Text Services Manager	7-17
Calling the Text Services Manager	7-17
Initializing as a TSM-Aware Application	7-18
Creating a TSM Document	7-18
Making Text Services Available to the User	7-20
Activating and Deactivating a TSM Document	7-20
Passing Events, Menu Selections, and Cursor Setting	7-21
Confirming Active Text Within a TSM Document	7-23
Deleting a TSM Document	7-24
Closing Down as a TSM-Aware Application	7-24
Requesting a Floating Input Window for Text Entry	7-24
Associating Input Methods With Scripts and Languages	7-25
Handling Text Service Apple Events	7-25
Receiving Text and Updating the Active Input Area	7-26
Converting Screen Position to Text Offset	7-29
Converting Text Offset to Screen Position	7-32
Showing or Hiding the Input Window	7-36
Direct Access to Text Service Components	7-36
Calling the Component Manager	7-36
Calling Text Service Components	7-37
Using the Text Services Manager (for Text Service Components)	7-37
Providing Menus and Icons	7-38
Providing a Text Service Component Menu	7-38
Providing Input Method Icons for the Keyboard Menu	7-39
Responding to Calls	7-40
Initiating a Text Service	7-41
Activating Text Service Component Windows	7-41
Responding to Events and Updating the Cursor and Menu	7-41
Confirming Active Text Input	7-42
Closing a Text Service	7-42
Identifying the Supported Scripts and Languages	7-42
Making Calls	7-44
Sending Apple Events to Client Applications	7-44
Opening Floating Utility Windows	7-48
Text Services Manager Reference	7-48
Text Services Manager Routines for Client Applications	7-48
Initializing and Closing as a TSM-Aware Application	7-49
Creating and Activating TSM Documents	7-50
Passing Events to Text Service Components	7-54
Passing Menu Selections and Cursor Setting	7-55
Confirming Active Input in a TSM Document	7-56
Making Text Services Available to the User	7-57
Requesting a Floating Input Window	7-61
Associating Scripts and Languages With Components	7-62

Apple Event Handlers Supplied by Client Applications	7-65
Creating and Updating an Active Input Area	7-68
Converting Global Coordinates to Text Offsets	7-72
Converting Text Offsets to Global Coordinates	7-74
Showing or Hiding the Floating Input Window	7-76
Text Services Manager Routines for Components	7-77
Sending Apple Events to a Client Application	7-77
Opening Floating Utility Windows	7-79
Text Service Component Routines	7-84
Providing a Text Service	7-84
Responding to Events and Updating the Cursor and Menu	7-87
Confirming Active Input in a TSM Document	7-89
Identifying the Supported Scripts and Languages	7-90
Summary of the Text Services Manager	7-92
Pascal Summary	7-92
Constants	7-92
Data Types	7-94
Text Services Manager Routines for Client Applications	7-95
Text Services Manager Routines for Components	7-97
Text Service Component Routines	7-97
C Summary	7-98
Constants	7-98
Data Types	7-101
Text Services Manager Routines for Client Applications	7-102
Text Services Manager Routines for Components	7-104
Text Service Component Routines	7-104
Assembly-Language Summary	7-105
Trap Macros	7-105
Result Codes	7-107

## Chapter 8

## Dictionary Manager 8-1

---

About Dictionaries for Input Methods	8-3
About the Dictionary Manager	8-4
The Structure of a Dictionary	8-5
Garbage Data	8-8
Dictionary Manager Limitations	8-10
Using the Dictionary Manager	8-11
Testing for the Presence of the Dictionary Manager	8-11
Making a Dictionary	8-11
Creating the File	8-12
Constructing the Dictionary	8-13
Accessing a Dictionary	8-13
Opening and Closing the Dictionary	8-13
Obtaining Information About the Dictionary	8-14

Locating Records in a Dictionary	8-15
Locating Records by Key	8-15
Locating Records by Index	8-17
Modifying a Dictionary	8-18
Compacting a Dictionary	8-20
Dictionary Manager Reference	8-20
Data Structures	8-20
Routines	8-20
Making a Dictionary	8-20
Accessing a Dictionary	8-22
Locating Records in a Dictionary	8-26
Modifying a Dictionary	8-30
Compacting a Dictionary	8-33
Summary of the Dictionary Manager	8-34
Pascal Summary	8-34
Constants	8-34
Data Types	8-34
Routines	8-35
C Summary	8-36
Constants	8-36
Data Types	8-37
Routines	8-37
Assembly-Language Summary	8-39
Trap Macros	8-39
Result Codes	8-39

## Appendix A

## Built-in Script Support A-1

---

The Roman Script System	A-4
The Standard Roman Character Set	A-4
Nonprinting Characters	A-6
Printing Characters	A-8
Variations in the Character Set	A-16
The U.S. Keyboard-Layout ('KCHR') Resource	A-19
Standard Sorting Routines	A-20
Diacritical Stripping and Case Conversion	A-23
U.S. International Resources and Keyboard Resources	A-23
WorldScript I	A-25
About WorldScript I	A-25
Shared Script Utilities and QuickDraw Patches	A-25
Table-Based Script Behavior	A-27
Contextual Formatting Routines	A-27
Flexible Dispatching Method	A-28
Initialization Sequence	A-28
How Calls Are Dispatched	A-29

Saving User Preferences	A-31
Replacing a Script Utility or QuickDraw Patch	A-32
Patching Script Utilities	A-33
Patching QuickDraw Routines	A-34
Issues in Designing a Script Utility or QuickDraw Patch	A-35
WorldScript II	A-36
About WorldScript II	A-36
Shared Script Utilities	A-37
Table-Based Script Behavior	A-38
Initialization Sequence	A-38
How Calls Are Dispatched	A-39

## Appendix B

## International Resources B-1

---

About the International Resources	B-4
What the International Resources Are	B-4
Script Codes and Resource ID Ranges	B-6
Using the International Resources	B-8
International Configuration Resource (Type 'itlc')	B-9
The ItlcRecord Data Type	B-10
Script-Sorting Resource (Type 'itlm')	B-12
International Bundle Resource (Type 'itlb')	B-17
The ItlbRecord Data Type	B-18
The ItlbExtRecord Data Type	B-20
Numeric-Format Resource (Type 'itl0')	B-22
The Intl0Rec Data Type	B-23
Long-Date-Format Resource (Type 'itl1')	B-28
The Intl1Rec Data Type	B-28
The Itl1ExtRec Data Type	B-31
String-Manipulation Resource (Type 'itl2')	B-34
Resource Header	B-35
The 'itl2' Sorting Hooks	B-37
The 'itl2' Tables	B-39
Script Run Table Format	B-40
Supplying Custom Sorting Routines	B-43
Supplying Custom Word-Break Tables	B-44
NBreakTable Format	B-44
How FindWordBreaks Uses the Break Table	B-49
Tokens Resource (Type 'itl4')	B-50
The NItl4Rec Data Type	B-51
The Token Table	B-53
The Extension-Fetching Routine	B-54
The Token-String Copy Routine	B-54
The Untoken Table	B-54
The Number Parts Table	B-55
The Whitespace Table	B-58

Encoding/Rendering Resource (Type 'itl5')	B-58
Resource Header	B-59
Tables for 1-Byte Script Systems	B-60
Script Configuration Table	B-60
Line-Layout Metamorphosis Table	B-63
Line-Layout Glyph-Properties Table	B-64
Character Expansion Table	B-64
Glyph-to-Character Table	B-65
Break-Table Directory	B-66
Script Run Tables	B-67
Kashida Preferences Table	B-68
Feature List Table	B-68
Reordering Table	B-68
Tables for 2-Byte Script Systems	B-68
Byte-Type Table	B-69
Character-Type Table	B-69
Transliteration Resource (Type 'trsl')	B-70
Resource Header	B-71
Rule-Based Format	B-72
Table-Based Format	B-73
Summary of the International Resources	B-74
Pascal Summary	B-74
Constants	B-74
Data Types	B-75
C Summary	B-80
Constants	B-80
Data Types	B-82

## Appendix C

## Keyboard Resources C-1

---

About Keyboards	C-4
About the Keyboard Resources	C-6
What the Keyboard Resources Are	C-7
Key Translation	C-8
Using the Keyboard Resources	C-10
Key-Map Resource (Type 'KMAP')	C-11
Apple Extended Keyboard	C-13
Reassigning Right-Hand Key Codes	C-14
Other Hardware Dependencies	C-14
Virtual Key Codes for Non-ADB Keyboards	C-15
Key-Remap Resource (Type 'itlk')	C-16
Keyboard-Layout Resource (Type 'KCHR')	C-18
Resource Format	C-18
The KeyTranslate Function and the Keyboard-Layout Resource	C-19



Special Uses for the KeyTranslate Function	C-22
Installing a Custom Keyboard-Layout Resource	C-22
Using KeyTranslate for Command-Key Equivalents	C-23
Keyboard Icon Family (Types 'kcs#', 'kcs4', 'kcs8')	C-25
Keyboard-Swap Resource (Type 'KSWP')	C-26
Key-Caps Resource (Type 'KCAP')	C-28
Resource Format	C-28
Key Caps Desk Accessory	C-32
Summary of the Keyboard Resources	C-35
Assembly-Language Summary	C-35
Global Variables	C-35

Appendix D	Renamed and Relocated Text Routines	D-1
------------	-------------------------------------	-----

---

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

---

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

---

