

Learn AppleScript

The Comprehensive Guide to Scripting and
Automation on Mac OS X, Third Edition



**Hamish Sanderson and
Hanaan Rosenthal**

Apress®

**Learn AppleScript: The Comprehensive Guide to Scripting and Automation on Mac OS X,
Third Edition**

Copyright © 2010 by Hamish Sanderson and Hanaan Rosenthal

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-4302-2361-0

ISBN-13 (electronic): 978-1-4302-2362-7

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Publisher and President: Paul Manning

Lead Editor: Douglas Pundick

Technical Reviewer: David A. Coyle

Editorial Board: Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Jonathan Gennick, Jonathan Hassell, Michelle Lowman, Matthew Moodie, Duncan Parkes, Jeffrey Pepper, Frank Pohlmann, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Coordinating Editor: Debra Kelly

Copy Editor: Bill McManus

Compositor: MacPS, LLC.

Indexer: BIM Indexing & Proofreading Services

Artist: April Milne

Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at www.apress.com/info/bulksales.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at www.apress.com.

Contents at a Glance

■ Contents at a Glance	iii
■ Contents	v
■ Foreword	xix
■ About the Authors	xxi
■ About the Technical Reviewer	xxii
■ Acknowledgments	xxiii
■ Introduction	xxiv
Part I: Welcome to AppleScript	1
■ Chapter 1: Introducing AppleScript	3
■ Chapter 2: AppleScript in Principle	15
■ Chapter 3: AppleScript in Practice	33
Part II: Understanding AppleScript	51
■ Chapter 4: Writing Scripts in AppleScript Editor	53
■ Chapter 5: Understanding How Application Scripting Works	77
■ Chapter 6: Learning to Work with AppleScript Objects	143
■ Chapter 7: Working with Text	163
■ Chapter 8: Working with Numbers	215
■ Chapter 9: Working with Dates	241
■ Chapter 10: Working with Lists and Records	267
■ Chapter 11: Storing Objects in Variables	293
■ Chapter 12: More on Commands	319
■ Chapter 13: More on Operators and Coercions	369
■ Chapter 14: Making Decisions Using Conditionals and Loops	389
■ Chapter 15: Making Decisions When Dealing with Errors	413
■ Chapter 16: Interacting with the User	437
■ Chapter 17: Working with Files	471

■ Chapter 18: Organizing Your Code with Handlers	505
■ Chapter 19: Organizing Your Code with Script Objects	541
Part III: Putting AppleScript to Work	585
■ Chapter 20: Scripting the File System	587
■ Chapter 21: Scripting Apple Applications.....	623
■ Chapter 22: Extending AppleScript with Scripting Additions	675
■ Chapter 23: AppleScript Amenities.....	711
■ Chapter 24: Scripting iWork and Office	743
■ Chapter 25: Scripting Data and Databases.....	793
■ Chapter 26: Scripting Adobe InDesign.....	817
■ Chapter 27: Interacting with the Unix Command Line.....	863
■ Chapter 28: Using Smile: The AppleScript Integrated Production Environment	897
■ Chapter 29: Tips and Techniques for Improving Your Scripts	927
■ Chapter 30: Creating Cocoa Applications with AppleScriptObjC	969
■ Index	1019

Contents

■ Contents at a Glance.....	iii
■ Contents.....	v
■ Foreword	xix
■ About the Authors.....	xxi
■ About the Technical Reviewer	xxii
■ Acknowledgments	xxiii
■ Introduction	xxiv
Part I: Welcome to AppleScript.....	1
■ Chapter 1: Introducing AppleScript	3
Automating with AppleScript: From Small Tasks to Large	3
Automating Small Tasks	3
Automating Large Tasks	4
When to Automate (or Not)?.....	5
The Golden Triangle: Scriptable Applications, the AppleScript Language, and the Scripts You Write	6
What Is a Scriptable Application?	6
What Is the AppleScript Language?	8
What Is a Script?	9
Your First AppleScript: Hello World!.....	10
Summary	13
■ Chapter 2: AppleScript in Principle	15
The Four Key Concepts of Application Scripting.....	16
How Applications Represent Information As Objects.....	16
How Applications Manipulate Objects Using Commands	18
How Applications Organize Their Objects into Object Models	19
How Applications Describe Their Objects and Commands in Dictionaries	21
The Four Key Concepts of the AppleScript Language.....	22
How AppleScript Represents Information As Objects.....	23
How AppleScript Manipulates Objects Using Commands, Operators, and Variables	24
How AppleScript Organizes Code Using Handlers and Script Objects	26

Tying It All Together: What Makes AppleScript Special	28
The English-like Syntax	28
Built-in Application Scripting Support	29
Attaching Scripts to Applications	29
The AppleScript Community	30
Summary	31
Chapter 3: AppleScript in Practice	33
The Project: Tidy Desktop	33
Planning the Script	34
Writing the Script	35
Creating the Folder Name	35
Interacting with the File System	39
Final Thoughts	46
Improving Reliability	46
Adding Features	47
Documenting Your Code	48
Summary	49
Part II: Understanding AppleScript	51
Chapter 4: Writing Scripts in AppleScript Editor	53
Working with Script Windows	53
Compiling a Script	54
Running and Stopping a Script	55
Recording a Script	55
Viewing the Result of Running a Script	56
Viewing the Events Sent by a Running Script	56
Adding a Description to a Script	57
Viewing Application Dictionaries	57
Other Useful AppleScript Editor Features	58
Using the Library Window	59
Navigating to a Handler in a Script	60
Viewing the Event Log History	61
Scripting the AppleScript Editor	61
Using Context Menu Scripts	62
Viewing the Bundle Contents Drawer	62
Setting the Script Formatting Preferences	63
Saving Scripts	65
Text-Based Scripts	66
Compiled Scripts	66
Script-Based Applications (Applets)	69
Other Tools for Writing and Running AppleScripts	72
The System-wide Script Menu	72
Third-Party AppleScript Editors	74
Developing GUI Applications	76
Command-Line Tools	76
Summary	76
Chapter 5: Understanding How Application Scripting Works	77
A Quick Application Scripting Example	78
Understanding Objects	79
How Objects Are Categorized by Class	79
Introducing Object Properties	81

Introducing Object Elements.....	84
Understanding Commands.....	86
Introducing Command Names	87
Introducing Command Parameters	87
Commands Must Be Targeted at Objects.....	89
Commands Can Return Results or Report Errors.....	90
Understanding the Application Object Model.....	91
How Objects Can Contain Other Objects.....	92
How AppleScript Refers to Application Objects	93
Exploring a Typical Object Model.....	98
Understanding Application Dictionaries.....	103
Introducing AppleScript Editor's Dictionary Viewer	104
How Classes Are Documented.....	107
How Commands Are Documented	118
Application Dictionaries Don't Tell You Everything!.....	120
More on <i>application</i> Objects	123
Creating <i>application</i> Objects.....	123
The Standard Properties of <i>application</i> Objects	126
How AppleScript Compiles <i>tell application ...</i> Blocks	127
More on Constructing References	128
Referring to Properties.....	129
Referring to Elements	129
Referring to Insertion Locations.....	138
Summary	139
Chapter 6: Learning to Work with AppleScript Objects	143
How AppleScript Represents Information As Objects	144
What Kinds of Objects Does AppleScript Provide?	144
Creating New Objects	145
Getting Information from Objects.....	145
How AppleScript Works with Objects	146
Manipulating Objects with Commands	146
Manipulating Objects with Operators	148
Coercing Objects.....	153
Storing Objects in Variables.....	154
Working with Boolean Objects.....	157
Boolean Operators	158
Boolean-Related Coercions.....	160
Summary	161
Chapter 7: Working with Text	163
Introducing Text.....	163
A Brief History of Text	164
Introducing Text in AppleScript 2.0.....	169
How to Write Literal Strings in AppleScript	170
The Properties of Text.....	174
The <code>class</code> Property	174
The <code>length</code> Property	174
The <code>quoted form</code> Property.....	174
The <code>id</code> Property	174
The Elements of Text	175
Characters	175

Words.....	178
Paragraphs.....	179
Text Ranges	179
Operators and Commands	181
Joining Strings.....	181
Comparing Strings.....	182
Checking for Strings Within Strings.....	184
Considering and Ignoring Attributes	185
The count Command.....	189
The offset Command.....	189
Coercing to and from Text	190
Working with Text Item Delimiters.....	192
Splitting Strings with Text Item Delimiters	192
Combining List Items into a Single String.....	194
Finding and Replacing Text	196
How AppleScript Text Works in Tiger	197
The Many Different Classes of Text	198
Working with the Various Text Classes.....	198
The ASCII number and ASCII character Commands.....	200
Example Projects	201
Defining a Find-and-Replace Command.....	202
Transferring Meeting Arrangements from Mail to iCal	204
Summary	213
Chapter 8: Working with Numbers	215
Introducing Integers and Reals.....	215
Operators and Commands	216
Math Operations	217
Comparison Operations	221
Introducing Operator Precedence	224
The <i>round</i> Command.....	225
The <i>random number</i> Command	227
Coercing to and from Numbers.....	229
Example Projects	231
Defining a Round-to-Decimal-Places Command	231
Blackjack!	233
Kitchen Timer.....	237
Summary	240
Chapter 9: Working with Dates.....	241
Introducing Dates.....	241
Understanding Date and Time Formats	242
Forming Dates On-the-Fly.....	244
Specifying a Time Relative to a Date	246
The Properties of Dates	247
The <i>class</i> Property	247
The <i>year</i> Property.....	247
The <i>month</i> Property	247
The <i>day</i> Property	248
The <i>weekday</i> Property.....	248
The <i>hours</i> Property	249
The <i>minutes</i> Property.....	249

The seconds Property	249
The time Property	249
The date string Property	250
The short date string Property	250
The time string Property	250
Setting Date Properties	250
Operators and Commands	252
Comparison Operators	252
Math Operators	254
The current date Command	257
The time to GMT Command	258
Example Projects	258
Deleting Old Files	259
Alarm Clock	263
Summary	266
Chapter 10: Working with Lists and Records	267
Introducing Lists and Records	267
What Is a List?	267
What Is a Record?	268
How Are Lists and Records Used in Scripts?	269
Working with Lists	269
The Properties of a List	270
The Elements of a List	271
Operators and Commands	274
Processing List Items with a Repeat Loop	279
Sorting Items in a List	281
Working with Records	282
Getting and Setting the Properties of a Record	284
Operators and Commands	286
Example Project: Measuring Word Frequency in TextEdit	287
Planning the Script	287
Developing the Script	288
Conclusion	291
Summary	292
Chapter 11: Storing Objects in Variables	293
Creating Variables	293
Declaring Your Variables	294
Choosing Your Variable Names	295
Working with Local Variables	299
Understanding Local Variable Scope	300
Passing Values to and from User-Defined Handlers	303
Working with Properties and Global Variables	305
Understanding Property and Global Variable Scope	305
How Properties (and Global Variables) Can Retain Values Between Runs	306
When to Use Properties and Global Variables	307
Using AppleScript's Built-in Variables	310
Variables Containing Predefined Values	311
Variables Containing Objects of Special Interest	313
Summary	318
Chapter 12: More on Commands	319

Understanding How and Where Commands Are Defined	319
Application-Defined Commands	319
Scripting Addition Commands	322
AppleScript Commands	323
User-Defined Commands	324
Picking the Right Target for Your Commands	325
The Five Standard AppleScript Commands	327
The <i>get</i> Command	327
The <i>set</i> Command	327
The <i>copy</i> Command	328
The <i>count</i> Command	330
The <i>run</i> Command	330
Commands That All Applications Understand	331
The <i>run</i> Command	331
The <i>launch</i> Command	332
The <i>open</i> Command	332
The <i>activate</i> Command	333
The <i>quit</i> Command	333
The Standard Suite of Application Commands	334
The <i>get</i> Command	334
The <i>set</i> Command	335
The <i>make</i> Command	336
The <i>exists</i> Command	338
The <i>count</i> Command	338
The <i>move</i> Command	339
The <i>duplicate</i> Command	340
The <i>delete</i> Command	341
The <i>print</i> Command	342
The <i>save</i> Command	343
The <i>close</i> Command	344
More on Working with <i>get</i> , <i>set</i> , and <i>copy</i> Commands	345
When Do Implicit <i>get</i> Commands Occur?	345
When Are Explicit <i>get</i> Commands Necessary?	347
Telling the Difference Between AppleScript and Application <i>set</i> Commands	350
Getting and Setting Multiple Values in a Single Statement	351
Using <i>copy</i> As an Application Command	352
Changing How Commands Are Handled	354
Considering and Ignoring Application Responses	354
Considering and Ignoring Other Attributes	357
Controlling the Timeout Delay for Long-Running Commands	357
Sending Commands to Remote Applications	359
Enabling Remote Apple Events	359
Talking to Remote Machines	361
Authenticating Access to Remote Machines	362
Compiling Scripts Offline	363
Using the choose remote application Command	364
Targeting Remote Applications by User and Process IDs	364
Launching Remote Applications	366
Summary	366
■ Chapter 13: More on Operators and Coercions	369
Understanding the <i>a reference to</i> Operator	369

Introducing Unit Type Conversions	373
A Summary of AppleScript Coercions	374
Coercing Class and Constant Names	375
Coercing Numbers and Text	376
Coercing File Objects	377
Coercing Lists	378
A Summary of AppleScript Operators	379
Boolean Logic Operators	379
The Concatenation Operator	379
Math Operators	380
Comparison Operators	381
Containment Operators	382
The Coercion Operator	383
The <i>a reference to</i> Operator	383
Operator Precedence in Detail	384
Understanding Precedence	385
Understanding Forms of Association	386
Understanding when You Should Use Parentheses	387
Summary	388
Chapter 14: Making Decisions Using Conditionals and Loops	389
Choosing Your Course with Conditional Blocks	389
Understanding the Basic Conditional Statement	390
The Different Flavors of Conditional Statements	391
Offering an Alternative Ending with the <i>else</i> Clause	391
Offering Multiple Choices with <i>else if</i> Clauses	392
Tips for Writing Conditional Statements	393
Running an Assembly Line with Repeat Loops	394
The Different Flavors of Repeat Loops	395
Repeating Forever	396
Repeating <i>n</i> Times	397
Repeating with Numbers in a Range	397
Repeating with Items in a List	400
Repeating While/Until a Condition Is Met	404
Tips for Writing Repeat Statements	406
Summary	410
Chapter 15: Making Decisions When Dealing with Errors	413
Understanding Compilation Errors	414
Simple Typos	414
Unbalanced Statements	414
Unbalanced Parentheses and Unescaped Quotes	414
Unrecognized Keywords	415
Reserved Word Mishaps	415
Invalid Date Literals	416
Understanding Runtime Errors	416
How Runtime Errors Work in AppleScript	416
Understanding the Psychology of an Error	419
Understanding the Anatomy of an Error	419
Trapping Runtime Errors	420
Generating Runtime Errors	425
Tips for Handling Runtime Errors	428
Understanding Common Error Numbers	431

Identifying Operating System Errors	431
Identifying Apple Event Errors	432
Identifying Application Scripting Errors	433
Identifying AppleScript Language Errors	434
Summary	435
Chapter 16: Interacting with the User	437
Creating Basic Dialog Boxes and Gathering Text Input.....	437
Introducing the display dialog Command	437
Introducing the display alert Command.....	447
Introducing the choose from list Command.....	449
Choosing Files, Folders, and Disks	454
Introducing the choose file Command.....	454
Introducing the choose file name Command	460
Introducing the choose folder Command.....	463
Choosing Other Types of Items	465
Introducing the choose application Command	465
Introducing the choose remote application Command	466
Introducing the choose URL Command.....	467
Introducing the choose color Command	468
Summary	470
Chapter 17 Working with Files	471
Identifying Files, Folders, and Disks	471
Using Path Strings	472
Using File System Objects	474
File URL Strings	482
Reading and Writing Files.....	483
Commands for Reading and Writing Files.....	483
Dealing with Text File Encodings	495
Summary	502
Chapter 18: Organizing Your Code with Handlers	505
What Are User-Defined Handlers?	505
Creating a Simple Handler	507
Using Parameters	508
What's the Result?	509
Using the <i>return</i> Statement.....	509
Working with Return Values	510
Can a Handler Return More Than One Result?	511
Specifying Handler Parameters	512
Using Positional Parameters.....	513
Using Labeled Parameters.....	515
Introducing Recursion.....	519
Using the <i>run</i> Handler in Scripts	520
I've Never Seen No <i>run</i> Handler!.....	520
When Should You Make the <i>run</i> Handler Explicit?.....	521
Working with Handlers in Applets.....	522
Using the Standard Event Handlers	522
Using User-Defined Handlers from Other Scripts	526
Tips for Designing and Using Handlers.....	526
Organizing Code.....	526
Reusing Code	527

Thinking Ahead	527
Thinking Small	527
Reorganizing Existing Code	528
Example Project: A Reusable Associative List	529
Designing the Associative List	530
Writing the Essential Handlers	531
Writing the Extra Handlers	534
Revising the Design	535
Conclusion	537
Summary	538
Chapter 19: Organizing Your Code with Script Objects	541
Introducing Script Objects	541
What Is a Script Library?	542
What Is Object-Oriented Programming?	543
Understanding Script Objects	543
Defining a Script Object in a Script	544
Loading and Storing Script Objects from Disk	545
How Variable Scope Works in Script Objects	546
Working with Script Libraries	548
What Advantages Do Script Libraries Provide?	549
Creating Your First Script Library	550
Where Should You Save Your Library?	550
What Are the Options for Loading Script Libraries?	552
Beginning Object-Oriented Programming	556
Your First Object: A Better Associative List	557
Extending Objects Through Inheritance	563
Creating Plug-and-Play Objects	573
Summary	583
Part III: Putting AppleScript to Work	585
Chapter 20: Scripting the File System	587
How the Finder Represents the Mac OS X File System	587
Understanding Containers	589
Understanding Files	592
Understanding Packages	594
Understanding the <i>computer-object</i> Class	594
How System Events Represents the Mac OS X File System	595
Understanding Files	597
Understanding Folders	598
Understanding Disks	599
Understanding System Events' <i>alias</i> Class	599
Other Classes of Interest	600
Working with Finder Items	602
Opening and Closing Items	602
Duplicating Files and Folders	603
Deleting Files and Folders	604
Moving Files and Folders	605
Checking if Items Exist	606
Making New Items	606
Sorting Lists of Finder References	607
Ejecting Disks	608

Selecting Items in a Finder Window	608
Converting Finder References to AppleScript Aliases.....	608
More on Working with Folders.....	610
Filtering the Content of Folders	610
Getting the Entire Contents of a Folder	611
Locating Important Files and Folders with Standard Additions	612
Avoiding Hardcoded Paths to the Startup Disk and Home Folder.....	613
Getting the Path to the Currently Running Script.....	614
Getting Paths to Resources in an Application Bundle.....	616
Mounting Volumes	618
Summary	621
Chapter 21: Scripting Apple Applications.....	623
Scripting iTunes.....	623
Understanding the iTunes Object Model.....	624
Working with Sources.....	626
Working with Tracks and Playlists.....	627
Example Project: My Themed Playlist.....	631
Scripting Mail.....	635
Working with Mailboxes	636
Working with Messages.....	636
Creating Outgoing Messages	641
Working with Mail Rules	642
Example Project: Building a Monthly Message Archive.....	646
Scripting iCal.....	650
Working with Calendars.....	652
Working with Events	653
Scripting Address Book	655
Working with People	656
Working with Groups	659
Working with Rollovers	661
Example Project: Looking Up Contact Information by Name.....	663
Scripting Automator.....	666
Using the Run AppleScript Action	666
Scripting the Automator Application.....	671
Summary	673
Chapter 22: Extending AppleScript with Scripting Additions	675
Introducing Scripting Additions and Scriptable Faceless Background Applications.....	675
Scripting Additions in Detail	677
Scripting Additions and the AppleScript Language	677
Installing Scripting Additions	678
Missing Additions and Garbled Scripts	679
Distributing Scripting Additions with Your Scripts.....	680
Understanding Scripting Addition Compatibility Issues.....	683
Understanding Scriptable Faceless Background Applications.....	688
Installing Scriptable FBAs	688
Distributing Scriptable FBAs with Your Scripts.....	688
Examples of AppleScript Extensions.....	689
Built-in Extensions.....	689
Third-Party Extensions.....	689
Using the Standard Additions Scripting Addition	690
Providing Audio Feedback	691

Pausing Your Scripts.....	692
Getting and Setting the System Volume	692
Getting System Information	693
Working with URLs.....	694
Using the Image Events Faceless Background Application	695
Processing Text with the Satimage Scripting Addition.....	697
Downloading and Installing the Satimage Scripting Addition.....	698
Performing Simple Find and Replace Tasks	698
Counting Occurrences of a Substring in a String.....	699
Finding and Replacing Substrings in a File.....	700
Searching and Replacing Across Many Files	701
Finding and Replacing Text with Regular Expressions	701
Batch-Changing File Names	707
Summary	710
Chapter 23: AppleScript Amenities.....	711
Using the Services Menu	711
AppleScript Editor Services	712
Creating Your Own Services	712
Scheduling Scripts with iCal	715
Triggering Scripts with Folder Actions.....	716
Defining Event Handlers for Folder Actions	717
Where Are Folder Action Scripts Stored?	717
Managing Your Folder Actions	717
Creating Your First Folder Action	719
Controlling Applications with GUI Scripting	722
Enabling GUI Scripting	723
The GUI Scripting Dictionary	723
GUI Scripting Object Structure	723
Basic GUI Scripting Examples	725
Using GUI Scripting Commands	726
Example Project: Searching in Preview	728
Understanding the Limitations of GUI Scripting.....	735
Working with the Clipboard	736
Getting Clipboard Data into AppleScript	736
Setting the Clipboard Data.....	737
Example Project: Creating PDF Cuttings from Selections in Preview	739
Summary	742
Chapter 24: Scripting iWork and Office	743
Scripting iWork	743
Scripting Pages.....	744
Scripting Keynote.....	751
Scripting Numbers	758
Scripting Microsoft Office	760
Working with Microsoft Entourage	761
Working with Excel	777
Working with Word	788
Where Next?	791
Summary	792
Chapter 25: Scripting Data and Databases.....	793
Automating FileMaker Pro with AppleScript.....	794

The FileMaker Object Model and the Commands You Can Use	795
Finding Data Quickly with the <i>whose</i> Clause	802
Running AppleScripts from Inside FileMaker Pro	806
Scripting Runtime Labs' MacSQL	807
Getting Connected	808
Using a Simple <i>select</i> Command	809
More on Result Sets	809
Some Useful Handlers	810
Database Events	812
Database Events Classes and Commands	812
Database Events Examples	813
Summary	814
Chapter 26: Scripting Adobe InDesign	817
Learning by Example	818
Understanding the InDesign Object Model	820
How InDesign Organizes Its Many Properties	820
How InDesign Manages Units of Measurement	823
How to Set Properties Safely when Making New Objects	824
Working with Documents	825
Working with Document Layers	829
Understanding InDesign References	831
Working with Text	831
Working with Text Stories	832
Working with References to Text	833
Working with Graphics	835
Drawing Lines	836
Example Project: An InDesign Scripts Launcher	837
Working with Tables	840
Importing Table Text from a Spreadsheet	841
Copying Formatting Between Cells	843
Working with Selections	845
Example Project: Captioning a Selected Graphic	846
Example Project: Cleaning Up Selected Text	848
Labeling Objects for Use in Workflow Scripts	850
Taking Advantage of AppleScript-Specific Preferences	852
Example Project: Replacing Tags with Images	854
Preparing InDesign	855
Processing the Tags	856
Preparing the Image Files	857
Placing the Graphics	858
Testing the Script	860
Using InDesign on a Server	860
Summary	861
Chapter 27: Interacting with the Unix Command Line	863
Why and When Are Shell Scripts Used in AppleScript?	863
Understanding Unix Scripting Concepts	864
The Unix Command Line and Mac Desktop Compared	864
How the Unix Command Line Works	865
Getting Help	869
Things to Watch Out For	870
Some Useful Unix Commands	872

Running Shell Scripts from AppleScript.....	873
Scripting the Terminal Application.....	873
Using the <i>do shell script</i> Command.....	875
Assembling Shell Script Strings.....	879
Passing Data to Standard Input.....	883
Running AppleScript from Shell Scripts.....	886
Using the <i>osascript</i> Command.....	886
Passing Parameters to the Script.....	887
Other Related Commands.....	888
Example Projects.....	888
Simple Text Processing Handlers.....	888
Changing File System Permissions.....	890
Analyzing PDF Files.....	893
Summary.....	896
■ Chapter 28: Using Smile: The AppleScript Integrated Production Environment.....	897
Introducing Smile.....	897
Introducing Smile's Integrated Engine Architecture.....	897
About Smile's Terminal.....	898
Why Should You Use Smile?.....	898
Downloading Smile.....	899
Using Smile.....	899
Using the AppleScript Command-Line Environment.....	900
Using Smile's Custom Dialog Boxes.....	901
Using Regular Expressions.....	903
Using the Graphic Engine.....	904
Using SmileLab.....	905
Example Project: Adding Text to a PDF Document.....	908
Preparing the PDF.....	908
Rolling Up the Scripts into a Graphical Interface.....	915
Further Exercises.....	925
Summary.....	926
■ Chapter 29: Tips and Techniques for Improving Your Scripts.....	927
Design, in a Nutshell.....	928
Taking Your Scripting Skills to the Next Level.....	928
How Much Design Do Your Scripts Need?.....	929
Starting Out Is Hard (But Gets Easier with Practice).....	931
How the Design-Driven Approach Changes the Way You Think About Code.....	933
Testing and Debugging Your Code.....	934
General Testing Tips and Techniques.....	935
Debugging in AppleScript Editor.....	938
Debugging in Script Debugger.....	941
Keeping Track of Changes.....	944
Improving the Performance of Your Code.....	948
The Dos and Don'ts of Optimization.....	948
Common AppleScript Optimizations.....	951
Advanced Optimization Techniques.....	956
Summary.....	966
■ Chapter 30: Creating Cocoa Applications with AppleScriptObjC.....	969
Example Project: HelloAppleScriptObjC.....	970
Creating the Project in Xcode.....	971

Adding the AppleScript Code in Xcode	973
Constructing the Interface in Interface Builder	973
Building and Testing the Application	979
Understanding How Cocoa Works.....	980
Understanding Outlets	980
Understanding Targets and Actions.....	980
How AppleScript Interacts with Cocoa Objects.....	982
Modifying HelloAppleScriptObjC to Set the Text Field	983
Updating the Code and Testing the Changes	983
What Is All That Code in the Default AppleScript File?	983
Example Project: PlaySound	985
Creating the Project and Adding the Code	985
Constructing the Interface	986
Building and Testing the Application	987
Understanding How It Works	988
Viewing the Cocoa Documentation in Xcode	990
How Cocoa Classes Are Documented	991
How Cocoa Methods Are Documented.....	993
Example Project: Snapshot.....	996
Creating the Project and Starting the Code	996
Constructing the Interface	996
Completing the Code.....	997
Understanding How It Works	999
Preparing the Text Fields Using an <i>awakeFromNib</i> Handler.....	1000
Building and Testing the Application	1001
Modifying Snapshot to Use Cocoa Bindings	1001
Adjusting the Code for Bindings	1001
Setting Up Bindings in Interface Builder.....	1002
Building and Testing the Application	1006
Modifying PlaySound to Use Bundled Sound Files.....	1006
Adding the Sound File Resources to the Project.....	1006
Adding the Code.....	1007
Adding the Pop-Up Button	1007
Building and Testing the Application	1007
Understanding How It Works	1008
Example Project: KitchenTimer.....	1009
Creating the Project and Starting the Code	1009
Constructing the Interface	1010
Completing the Code.....	1013
Extra Exercises	1015
Where Next?	1016
Summary	1017
Conclusion	1018
■ Index.....	1019

Foreword

One of the inherent tensions in product design is the balance between generality and specificity. Products are designed to appeal to a wide range of users, yet each individual has specific needs and problems that cannot be anticipated.

Custom products are usually much more expensive than generic ones. This is true for clothing, automobiles, medicine, and of course software. Custom products are also more difficult to create. Even worse, a person's needs may change over time, so something that worked when it was purchased may not be as useful in another time or situation.

Unlike physical products, software is infinitely malleable. In theory, at least, it can do (almost) anything you want and be specialized to your needs. In practice, software designers have to anticipate what users are most likely to want. Building in every possible option would make the software very powerful but completely unusable. With this in mind, Mac software developers tend to focus on simplicity and elegance. This is a cultural value that has grown up with the Mac OS and now permeates every aspect of the Mac experience. The idea is to carefully design just the right product while avoiding unnecessary features.

But what if the end result is not quite right for you? When it comes to software, the human usually learns to adapt to fit the software, rather than the other way around. It's as if the tailor adjusts your body to fit the suit, rather than the suit to fit you. But there is another solution, which is to make the software itself more customizable. Again, it's not possible to anticipate everything that anyone would want. This is a difficult problem, then, to build software applications that are simple and elegant, yet infinitely customizable to suit unknown needs.

This is the vision behind AppleScript: to enable users to extend, combine, and customize software to suit their own specific needs. AppleScript is not just a scripting language. It is an approach to customizable and extensible software development, which has been adopted by hundreds of applications on the Mac. AppleScript is designed to create a network effect in software, where the more applications that use AppleScript the better it is for everyone. This is a grand vision, but the reality is not so simple. Scripting and customizing applications is not something that everyone does every day. It requires some technical knowledge and practical examples to follow.

This book is an essential guide to AppleScript and related technologies. The key value of the book is that it integrates a clear presentation of the vision and purpose of AppleScript with the mundane technical details of writing scripts and getting useful work done. When talking about scripting, it's easy to get lost in the details. When writing a foreword, like this, it's easy for me to present a grand vision that seems appealing. The benefit of this book is that it combines and connects the vision with the practical reality.

Dr. William Cook
Associate Professor
Department of Computer Science
University of Texas at Austin
AppleScript Architect and Engineering Manager
Apple Computer, 1991–1993

About the Authors

Hamish Sanderson is a 10-year veteran of AppleScript and a professional automation developer living and working in the UK. Over the years he has contributed a range of useful libraries and utilities to the AppleScript community and pops up with some regularity on the various AppleScript mailing lists. He is also the author of the highly regarded appscript library, which brings AppleScript-style application scripting support to the Python, Ruby, and Objective-C languages on Mac OS X. He can be reached at hhas@users.sourceforge.net.



Hannan Rosenthal is the founder of Custom Flow Solutions LLC. He started his media automation consulting firm in 1990, whose clients are now The New York Times, Thomson Reuters, Associated Press, and Fidelity Investments. Since then he helped reshape the workflow of numerous publishing companies creating systems that save their owners millions of dollars and thousands of hours annually.

Hanaan is a speaker in MacWorld Expo and Seybold Seminars. Hanaan and his wife, Johanne, home-school their two children and restoring their solar-powered Victorian home in Providence, RI. In his spare time, Hanaan likes to pour concrete kitchens, sculpt, photograph, write and generally get into trouble in any way possible. He can be reached at applescript@hanaan.com.

About the Technical Reviewer

David A. Coyle earned his Ph.D. in thermochronology in 1992. Research in thermal modeling and crustal evolution at the Max-Planck Institute for Nuclear Physics (Heidelberg, Germany) lead to a career in software development. David has been concentrating on Apple technologies since 1997 and is a certified Apple System Administrator.

Acknowledgments

We would like to thank the many people who have worked long and hard to make this book possible.

Many thanks to our guest authors—Emmanuel Levy, Harald Monihart, Ian Piper, Shane Stanley, Barry Wainwright, and Craig Williams—for sharing their knowledge and experience with us.

Many thanks to David Coyle for helping to get the technical details right and to Bill McManus for ensuring all of the other words made sense too.

Many thanks too to Rob and Mark for their feedback on draft chapters and to Debra, Douglas, Kylie, Clay, and the rest of Apress for supporting this project through a long and sometimes arduous development process.

Lastly, many thanks to Dr William Cook for gracing us with his insights into the philosophy and goals of AppleScript and to all of the AppleScript engineers, application developers, and community members for a network effect where the copious sharing of knowledge and enthusiasm is every bit as important as the myriad interactions of bits and bytes.

Introduction

Welcome to the third edition of *Learn AppleScript: The Comprehensive Guide to Scripting and Automation on Mac OS X*. This book will teach you the principles and practices of controlling the many “AppleScriptable” Mac OS X applications using the AppleScript scripting language included in Mac OS X.

The title of this book represents a challenge: just how do you create a comprehensive guide to a subject as broad and deep as automating your Mac with AppleScript?

While we do cover every aspect of the AppleScript language in extensive detail, obviously we cannot hope to cover every last scriptable application on the market today. Instead, we will explain the core principles by which all scriptable applications operate, before exposing you to a broad range of very different applications—from simple and familiar tools such as TextEdit and Finder, through popular lifestyle applications such as Mail and iTunes, and on to professional productivity programs such as Microsoft Office and Adobe InDesign.

Although you may not use all of these applications in your daily routine, we hope that you will take the time to explore them here. Exploring the wide range of applications covered here will help to familiarize you with the many similarities (and differences) found among scriptable applications today—knowledge that will provide essential guidance and inspiration as you go on to learn and use other scriptable applications for yourself.

While this book is designed for Mac users who have never scripted before, we have ensured that there is plenty of material here to engage more experienced AppleScripters, as well as professional programmers coming to AppleScript from other languages. So without further ado, let’s look at how you can get the most out of this book whatever your background is.

If You Have Never Scripted Before...

If you have never scripted before, don’t worry! This book was designed with you foremost in mind.

Part 1 of the book will start you off with a very gentle introduction to AppleScript and application scripting concepts and techniques. Learning to script is not a five-minute task; there is a lot of information to absorb, and it takes time and practice to master. To help you get to grips with these subjects, Chapter 2 divides them into eight “Key Concepts”—four for the AppleScript language, four for application scripting—to provide you with a roadmap of what you will learn

over the course of the book and help you to fit the pieces together as you go. Chapter 3 then leads you through the process of creating your first practical AppleScript.

Part 2 begins with an introduction to Mac OS X's AppleScript Editor, where you will write and test the majority of your scripts throughout this book. Next, Chapter 5 takes you on a grand tour of the four key concepts behind application scripting, helping you develop a robust understanding of this incredibly powerful technology in preparation for Part 3 of the book. Chapters 6 to 17 then guide you through the first three key concepts that underpin the AppleScript language itself, equipping you with all the knowledge you need to start writing scripts of your own. Chapters 18 and 19 cover the fourth key concept behind AppleScript, although as this material is more advanced you might wish to skip it for now and come back to it later when you are ready to learn more.

Part 3 of the book provides you with lots of opportunity to put your newfound knowledge to practical use, as you explore a wide range of popular scriptable Mac OS X applications in Chapters 20 to 23 and learn how to extend the functionality of the AppleScript language through the use of scripting additions.

As your confidence grows, you might like to tackle some of the larger, more advanced applications covered in Chapters 24 to 28. Or you might wish to return to Chapters 18 and 19 to complete your knowledge of AppleScript's most advanced language features before Chapter 30 launches you into the exciting world of GUI application development with Snow Leopard's new AppleScriptObjC bridge.

If You Are Already an AppleScript User...

If you are an existing AppleScript user keen to expand your knowledge and hone your skills, you will find lots of valuable information throughout this book, particularly in Parts 2 and 3. I recommend you start with a quick read through Part 1, just to familiarize yourself with the planned structure of the book, before diving into the areas of Parts 2 and 3 that interest you the most.

As with previous editions, Part 2 provides an extensive reference to every facet of the AppleScript language itself, while Part 3 gets down and dirty with the practical business of automating your desktop. However, this new edition expands both the breadth and depth of our coverage of AppleScript and application scripting with several hundred pages of completely new material!

In addition to ensuring a robust understanding of basic AppleScript features—objects, commands, operators, variables, and flow control statements—we dig deeply into the more advanced and obscure corners of AppleScript. Chapters 7 and 17 include detailed discussion of Unicode text and text file encodings—essential background knowledge for reading and writing text files in AppleScript. Chapter 17 clarifies the sometimes confusing world of AppleScript's alias, file, and POSIX file classes, explaining just how they work and how to use them effectively.

Chapters 18 and 19 provide detailed coverage of handlers and script objects—powerful features for organizing large, sophisticated scripts. Along the way, we show you how to package your handlers as reusable libraries for easy sharing between many scripts and provide an AppleScript-friendly introduction to object-oriented programming as you learn how to design and use your own “classes” of objects. With the introduction of Snow Leopard's exciting new AppleScriptObjC bridge, these features and techniques are more important to AppleScript developers than ever before.

Chapter 22 will help you navigate the practical issues involved in installing and using scripting additions across a range of Mac OS X architectures: PowerPC and Intel, 32- and 64-bit. Guest author Ian Piper also contributes an introduction to advanced text processing with the popular Satimage scripting addition.

As before, we cover a good range of popular Mac OS X applications—including the Finder, Mail, iCal, and the ever-present iTunes—giving you plenty of opportunity to hone your automation skills. In addition, two new guest chapters provide in-depth coverage of a number of important productivity applications. Chapter 24, contributed by Harald Monihart and Barry Wainwright, explores the iWork and Microsoft Office application suites, while Chapter 26 provides an extensive introduction to one of the great heavyweights of Mac OS X automation, Adobe InDesign, under the guidance of InDesign scripting expert, Shane Stanley.

Chapter 27 explains the ins and outs of integrating the AppleScript language with the powerful, if cryptic, Unix command line that lies just beneath the surface of the Mac OS X operating system, while Chapter 28 explores the highly versatile Satimage Smile platform in the company of guest author Emmanuel Levy.

In Chapter 29, we step back from the day-to-day practicalities of AppleScripting to examine the bigger picture behind scripting and programming in general. While this chapter is by no means required reading, we hope that it will provide valuable inspiration and practical knowledge that will aid you as you look to design, debug, and optimize your scripts more effectively.

Finally, in Chapter 30, guest author Craig Williams introduces you to one of the most exciting AppleScript developments of recent years: Snow Leopard's new AppleScriptObjC bridge. AppleScripters can now directly harness the enormous power of Mac OS X's Cocoa programming frameworks to develop highly polished and capable GUI applications of their own.

If You Are an Experienced Programmer...

If you are an experienced programmer looking to develop some AppleScript skills, you will find this book to be a valuable reference guide.

While the AppleScript language itself is surprisingly small and simple, the English-like syntax can take some getting used to if you are more familiar with dot-style languages such as Python and Ruby or curly-brace languages such as Objective-C and Java. Remember, AppleScript syntax is designed to be easily read by nonprogrammers, so the trade-offs it makes are different from those made by these other languages, but with a little time and practice I'm sure you'll get the hang of it too.

The real challenge will be in forming a full and accurate understanding of how application scripting works. AppleScript's command and reference syntax may resemble that of object-oriented programming—but don't be fooled by appearances! AppleScript's conceptual roots lie as much in procedural and relational database programming as in the more familiar world of OOP. (To use a rough analogy, think of sending Apple events as a bit like sending XPath queries over XML-RPC.) While we may not employ such technical terms throughout this book, the information you need to make sense of this powerful, if unusual, technology is all in there.

You may find the pacing a little slow for your taste and the detailed discussion of basic programming language concepts such as variables, values, and flow control to be somewhat redundant at your level. However, all the essential information you require is in here: the features and capabilities of the AppleScript language, and the principles and practices of application scripting. Nor do we shy away from "advanced" AppleScript topics such as module management and object-oriented programming with script objects, or essential optimization techniques for getting the best performance out of your application automation code.

We recommend you quickly skim the introductory material in Part 1 before moving on to the real meat in Parts 2 and 3. You'll find Chapters 6 and 12 especially valuable in understanding how application scripting works, while Part 3 provides lots of hands-on experience in working with a wide range of scriptable applications and explains how to integrate AppleScript with the Unix command line and Objective-C/Cocoa environments.

Resources

Here are some links to useful online AppleScript-related resources:

- The Apple web site provides several AppleScript-related mailing lists (<http://www.lists.apple.com/mailman/listinfo>), including the AppleScript-Users mailing list, which is a great place to meet other AppleScript users and get your questions answered. An AppleScript bulletin board is also available at <http://discussions.apple.com> (you'll find it in the Mac OS X Technologies section).
- The Apple web site's developer section (<http://developer.apple.com>) includes a range of AppleScript-related material aimed at intermediate and advanced AppleScript users and scriptable application developers, such as the AppleScript Language Guide reference documentation.
- MacScripter (<http://www.macscripter.net>) is a terrific community-driven web site where you can find scripting additions, AppleScript code shared by other scripters, a BBS, and many other AppleScript-related resources.
- The Mac OS X Automation web site (<http://www.macosxautomation.com>) provides useful information on AppleScript and other automation technologies found in Mac OS X. The AppleScript section includes links to a range of other AppleScript-related web sites.
- More-advanced readers might also wish to check out Dr. William Cook's HOPL III conference paper on the early history of AppleScript. Dr. Cook's paper provides many fascinating insights into the philosophy and workings of the AppleScript language and application scripting technologies and is available from his web site (<http://userweb.cs.utexas.edu/users/wcook>) as a downloadable PDF.

Many other AppleScript-related web sites exist that I haven't included here. Some are given in later chapters, others can be found via the previous links provided or by searching online.

Formatting Conventions Used in This Book

To aid readability, we have used a few simple formatting conventions in this book:

- The `↵` symbol is used in example scripts to indicate where a long line of code has been wrapped to fit the printed page. AppleScript also uses this symbol to indicate line continuations, so you can type it as-is if you wish, but its position may change when the script is compiled. Alternatively, you can omit both this symbol and the line break that follows it and allow the entire line to wrap naturally within the AppleScript Editor window.
- The `-->` character sequence at the start of a line is used to indicate a typical value returned by a script when it is run and is not . The double-hyphen sequence (`--`) is also used by AppleScript to indicate a single-line comment, so it doesn't matter if you type this line when entering a script into AppleScript Editor (though see the next point).
- Where a return value indicated by the `-->` character sequence is too long to fit on a single line, it may be spread across subsequent lines like this:

```
--> {application file "Address Book.app" of folder "Applications" of
      startup disk of application "Finder", application file "Automator.app"
      of folder "Applications" of startup disk of application "Finder", ...}
```

If you choose to type this text into AppleScript Editor, make sure you remove all of the line breaks so that it appears as a single line, otherwise the script will not compile correctly.

- We regularly use the phrase “application object” to mean any object that is defined by a scriptable application—a folder object, a playlist object, and so on. Most scriptable applications also define a top-level object whose class is application, so when we discuss these objects in particular, the word “application” is shown in monospace text to make the distinction clear; for example, “the Finder’s application object.”

Errata

We have made every effort to ensure that there are no errors in the text or code. However, mistakes happen, and we recognize the need to keep you informed of any mistakes as they’re discovered and corrected. An errata sheet will be made available on the book’s main page at <http://www.apress.com>. If you find an error that hasn’t already been reported, please let us know.

The scriptable applications used in this book were the latest versions available at the time of writing, though newer releases may appear over time. While scripts written for one version of an application will usually work with newer versions, this is not always the case. If you discover a script no longer works due to non-backwards-compatible changes in the software used, please let us know and we will try to address such issues where possible.

Part



Welcome to AppleScript

Chapter 1

Introducing AppleScript

AppleScript is a scripting language for automating your Mac. With AppleScript's help, many tasks that you would normally perform with the mouse and keyboard can be turned over to your Mac to deal with, freeing you up to get on with more interesting or important things.

What is AppleScript good for? AppleScript's greatest strength is its ability to communicate with hundreds, if not thousands, of Mac OS X applications. This makes it ideal for all kinds of application scripting tasks—anything from simple, one-off scripts that you run once to perform a particular task, up to large, complex workflows that will perform vital functions for months or even years to come. In fact, most of the power of AppleScript comes not from the language itself, but from the applications it talks to.

What makes AppleScripting worthwhile? Although many of the tasks you carry out on your Macintosh are interesting and unique, many other, day-to-day tasks are boring and repetitive, requiring you to perform the same sequence of actions over and over again. If you write some scripts to take care of the latter, the time you save can be spent on more of the former.

Automating with AppleScript: From Small Tasks to Large

AppleScript is great for creating small, simple scripts, each of which performs a single task many times over—providing significant, immediate benefits to even the most casual of users. In the hands of an experienced developer, AppleScript can also scale to large and complex tasks, to assemble a whole series of professional Mac applications into powerful and sophisticated workflows.

Let's consider an example of each.

Automating Small Tasks

Creating small, simple scripts is the ideal way for you to start to become familiar with AppleScript. AppleScript lets you easily create a script, test it, save it as a file or