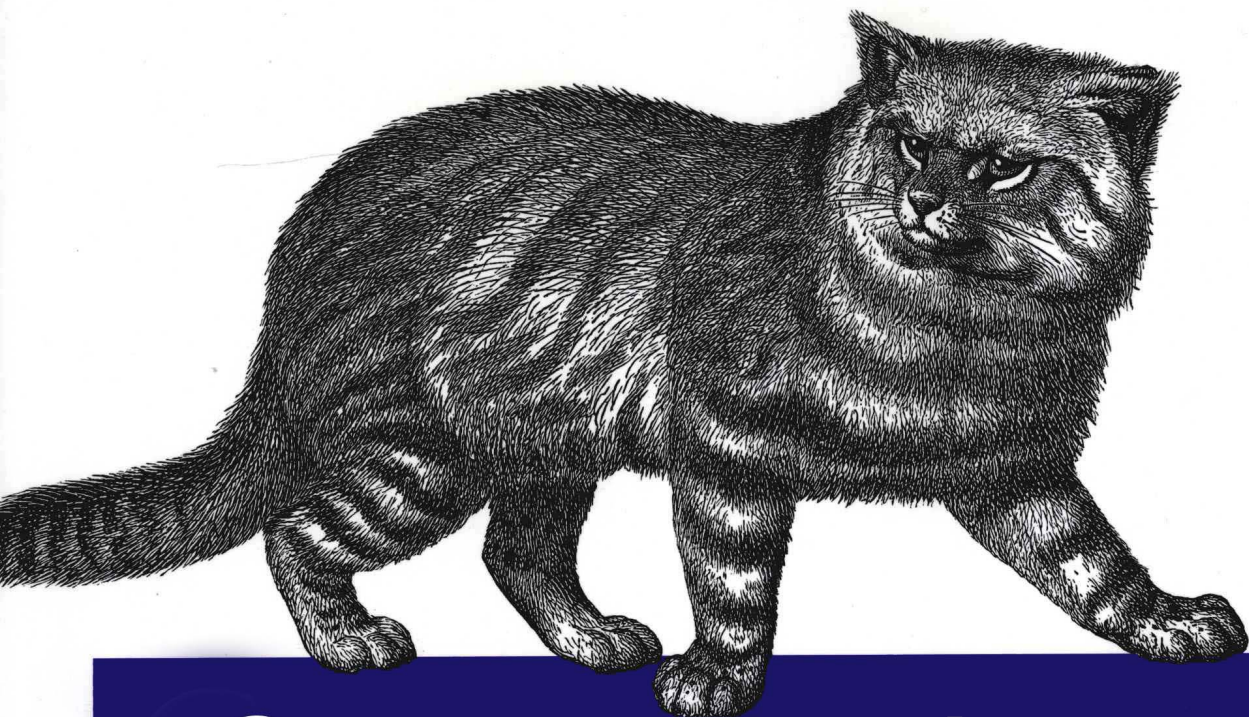


*Cocoa*与Objective-C: 构建与运行 (影印版)



Cocoa and Objecuve-C

Up and Running

O'REILLY®

東南大學出版社

Scott Stevenson 著

Cocoa与Objective-C: 构建与运行 (影印版)

Cocoa and Objective-C: Up and Running

Scott Stevenson

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

东南大学出版社

图书在版编目 (CIP) 数据

Cocoa 与 Objective-C: 构建与运行: 英文 / (美) 史蒂文森 (Stevenson, S.) 著. — 影印本. — 南京: 东南大学出版社, 2010.10

书名原文: Cocoa and Object-C: Up and Running

ISBN 978-7-5641-2413-7

I. ① C… II. ① 史… III. ① 操作系统, Mac OS X — 程序设计 — 英文 IV. ① TP316.89 ② TP312

中国版本图书馆 CIP 数据核字 (2010) 第 169657 号

江苏省版权局著作权合同登记

图字: 10-2010-283 号

©2010 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2010. Authorized reprint of the original English edition, 2010 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2010。

英文影印版由东南大学出版社出版 2010。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

Cocoa 与 Object-C: 构建与运行 (影印版)

出版发行: 东南大学出版社

地 址: 南京四牌楼 2 号 邮编: 210096

出 版 人: 江 汉

网 址: <http://www.seupress.com>

电子邮件: press@seu.edu.cn

印 刷: 扬中市印刷有限公司

开 本: 787 毫米 × 980 毫米 16 开本

印 张: 26.25 印张

字 数: 514 千字

版 次: 2010 年 10 月第 1 版

印 次: 2010 年 10 月第 1 次印刷

书 号: ISBN 978-7-5641-2413-7

印 数: 1~1600 册

定 价: 52.00 元 (册)

本社图书若有印装质量问题, 请直接与读者服务部联系。电话 (传真): 025-83792328

Preface

I'm not sure if this is the first book you've picked up to learn Cocoa, but I think it's the one that will get you started writing apps. I started teaching Cocoa and Objective-C in 2004, and I have worked with a lot of people who wanted to learn how to write software so that they could get their ideas onto the screen. But there's a problem.

There are two kinds of people who want to learn programming. Those in the first group are wired for the algorithmic mindset; they're interested in data and the inner workings of things for their own sake. When they see a dog catch a Frisbee, they think of the calculations the dog does to catch it. Programming is a natural extension of this mindset. My guess is around five percent of the population is actually built this way.

The second (much larger) group has ideas for software that they desperately want to make real. They're often graphic or interaction designers. You probably have ideas about a fantastic Mac, iPhone, or iPad app that you want to create, but you don't have a million dollars to hire an engineering staff. It's very likely that you even like programming and data in addition to several other interests, but you don't see everything in terms of algorithms.

The problem is that most technical books are written by and for people in the first group. That means most of the material is being created for those who need the least help. As a result, a lot of books on programming go unread. In fact, there's this unspoken honor given to anyone who actually finishes reading one: "Wow, he must be really motivated."

My problem with this is that there are a lot of people with great ideas in the second group. Many of my favorite Mac and iPhone apps today come from developers without a formal computer science background, most likely because they bring different experiences into the mix. I want to encourage more of this. Fortunately, the good folks at O'Reilly agree with me.

So here's the deal. I wanted to write this book because I want to help you learn how to write Mac, iPhone, and iPad apps. I want you to read it so that I get to use whatever software you end up creating.

This book is made to be accessible to new programmers, but it's not watered down. You're learning to use the same things the professionals use. My job is to make sure that each page says something useful. For each paragraph, I've asked myself, "Does this help you write your app?" Anything that didn't meet that standard got cut. But I haven't sold you short; if there's something you need to know to be a good Mac programmer, I've at least told you about it. However, I haven't spent time on minutiae that don't matter for Cocoa.

The content of this book is based on Cocoa tutorials I wrote between 2003 and 2009. Many of these were published at my personal site, Theocacao (<http://theocacao.com>), and some of the longer ones were published at Cocoa Dev Central (<http://cocoadevcentral.com>), a site I didn't originally create but have run since 2004. I've refined the tutorials based on a one-on-one mentoring program that I ran over the same period of time. You get the benefit of all those efforts in a single condensed book.

Your job is to go write world-class Mac, iPhone, and iPad apps, and to tell everyone else how great Cocoa is. Let's get started.

Who This Book Is For

This book is for people who want to learn to make great Cocoa apps. I don't assume that you already know how to program, or anything about Objective-C or C. You do need to own an Intel-based Mac running Mac OS X 10.6 Snow Leopard, and you must know how to install software, launch apps, edit and save files, and so on. Essentially, you have to know how to use a Mac.

If you know any computer languages at all (even HTML), things will make more sense from the start. If not, you'll still be able to make it through this book, but you may find some parts challenging. Even though Cocoa makes many common tasks easy, your brain has to adjust to the basic concepts of programming.

To be clear, though, this book is not exclusively for novice programmers. Depending on your experience level, you can skip a few of the chapters that are designed for beginners and jump right to the parts that are relevant to your experience.

If you already know C but haven't done object-oriented programming, start with Chapter 1, which will walk you through creating a basic Cocoa application. Then move onto Chapter 4, which introduces object-oriented concepts.

If you know C and at least one object-oriented language (such as Java, Ruby, or C++), you can start with Chapter 1 for the basic orientation, and then jump ahead to Chapter 5, which introduces Objective-C.

How This Book Is Organized

The chapters in this book are organized as follows:

Chapter 1

To build Cocoa apps, you'll need to know your way around Xcode. Although we'll get into Xcode more deeply later, this chapter gives you a quick tour.

Chapter 2

Before you can start programming in Objective-C (the native programming language used with Cocoa), you'll need a background in programming as well as in the C language. This chapter gets you started with the basics of programming in C.

Chapter 3

A programming language spends all its time moving things around in memory. This chapter explains how C manages memory and also explains pointers, which let you work directly with memory locations. Although you won't need all the low-level memory manipulation that C is capable of, an understanding of it will help you better understand Objective-C.

Chapter 4

Here's where we take a detour from the C language and get into the object-oriented world. In this chapter, you'll learn about classes, inheritance, objects, and more.

Chapter 5

Now that you have a basic understanding of object-oriented concepts, it's time to move on to Objective-C. This chapter explains Objective-C's syntax for calling methods, defining classes, and creating objects.

Chapter 6

Before you can get into Cocoa, there are a few more things you need to learn about Objective-C. This chapter introduces some intermediate Objective-C concepts, including memory management, categories, selectors, and more.

Chapter 7

Although you can (and sometimes will) use standard C types in your Cocoa apps, Objective-C offers a rich set of classes for working with primitive values, such as integers, floating-point numbers, and strings. This chapter shows you how to use these value classes.

Chapter 8

Cocoa's AppKit user interface layer allows you to create applications with rich user interfaces. This chapter prepares you to work with the built-in controls and connect them with actions you define in your code.

Chapter 9

Model-View-Controller is the mindset that guides the way you'll put your Cocoa apps together. In this chapter, you'll learn how to write code that coordinates your data and user interfaces.

Chapter 10

Cocoa includes a rich set of classes for displaying graphics in your apps. In this chapter, you'll learn how to work with shapes, images, gradients, and more.

Chapter 11

Now that you’ve read through the book, you’re ready to write some apps. This short chapter gives you a few last pointers to help you on your way.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, and file extensions.

Constant width

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

Constant width bold

Shows commands or other text that should be typed literally by the user.

Constant width italic

Shows text that should be replaced with user-supplied values or by values determined by context.



This icon signifies a tip, suggestion, or general note.



This icon indicates a warning or caution.

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you’re reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O’Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product’s documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: “*Cocoa and Objective-C: Up and Running* by Scott Stevenson. Copyright 2010 Scott Stevenson, 978-0-596-80479-4.”

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

Safari® Books Online



Safari Books Online is an on-demand digital library that lets you easily search over 7,500 technology and creative reference books and videos to find the answers you need quickly.

With a subscription, you can read any page and watch any video from our library online. Read books on your cell phone and mobile devices. Access new titles before they are available for print, and get exclusive access to manuscripts in development and post feedback for the authors. Copy and paste code samples, organize your favorites, download chapters, bookmark key sections, create notes, print out pages, and benefit from tons of other time-saving features.

O'Reilly Media has uploaded this book to the Safari Books Online service. To have full digital access to this book and others on similar topics from O'Reilly and other publishers, sign up for free at <http://my.safaribooksonline.com>.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

<http://www.oreilly.com/catalog/9780596804794/>

A full site dedicated to the book is available from the author, and may include additional examples and announcements about Cocoa user groups and other information you may find useful as a developer. This book site is at:

<http://cocoabook.com>

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

You can email the author directly at:

cocoahelp@me.com

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our website at:

<http://www.oreilly.com>

Acknowledgments

My education didn't follow the normal path. I taught myself how to program and learned how to run a business mostly by trial and error. In the end, I think this is the only option that would have worked for me, but it was possible only because of my extremely patient and understanding family: my mom, Peggy; my dad, Alan; and my sister, Jamie. You would not be reading this now if it was not for their support.

There are so many people that have helped me in my work life, but there are a few that have had a direct impact on this book.

Michael Lopp and Angela Muller were ongoing sources of encouragement and inspiration. This book first came into being over lunch when Michael mentioned something along the lines of "everyone wants an animal on the cover of their book," referring to the iconic O'Reilly covers. By incredible coincidence, Brian from O'Reilly emailed me about two weeks later. Had Michael not made that comment, this might not have happened.

I had world-class tech reviewers for this project: Joar Wingfors, Michael Jurewitz, Rob Rhyne, and Tim Triemstra. Joar, who I originally met through the tutorials I posted online, tirelessly reviewed an ever-changing book, provided a wealth of essential comments and suggestions, and even helped me fix some bugs in the code. He's one of the most talented engineers I know, and I am thrilled to have been able to get his help on this.

Though he wasn't involved in this project, I owe a lot to John Mora. He has an ability to look at an impossibly large task in the face and just do it, despite the usual doubts about whether you know enough or have enough time. It is one of the most impressive traits I know of, and John has it in spades. Thankfully, I think some of that rubbed off on me over the 15 years I've known him.

Thanks to Kip Krueger for patiently helping me figure out the low-level details of memory and offering pointers when I was learning C. His mentoring is the reason I'm able to teach these topics to you now.

When I first started learning Cocoa, there were only two books out on the topic. I chose Aaron Hillegass's book *Cocoa Programming for Mac OS X* (Addison-Wesley). Not only was it the first Mac programming book I read, it was the first technical book I actually enjoyed. His book proved to me that you could write in a direct and personal style and still be taken seriously. Having spent time with him in person, I can also add that he's a great guy. There's no doubt that his influence is weaved into the tutorials I've written.

Brian Jepson is simply a superhero among editors. I'm not sure there's enough space to list all of the things he did to make sure this project was a success, but I certainly could not have done it without him. For any part of the book that you really like, there's a good chance he had a hand in it. Follow him on Twitter: [@bjepson](#).

Finally, to the thousands of people who have emailed me over the years with questions or suggestions about tutorials, I owe a lot to you as well. You helped me refine everything that ended up in this book, and by extension, helped a new generation of Cocoa developers learn how to program.

This book is dedicated to Gina and Ilya, who both taught me that life's too short to not be spent with the people you want to be with.

Thank you.

Table of Contents

Preface	xi
 1. Setup and First Run	 1
Download and Install Xcode	1
Your First Application	4
Create the Interface	7
Run the Finished Application	11
 2. Thinking in Code: Basic C	 15
How Code Works	15
How to Format Code	17
Variables	20
Types	21
Constants	23
Enumerated Types	24
Typedefs	24
Functions	24
Declaring Functions	28
Example: FirstProgram	28
Displaying Values on the Command Line	29
Compile and Run the Example	32
Scope	34
Static Variables	35
Conditionals	36
Example: ShoppingTrip	39
Compile and Run	41
Wrap Up	42
 3. Memory and Pointers: Advanced C	 43
Arrays	43
Loops	45

Text Strings	47
Multidimensional Arrays	48
Pointers	49
The Purpose of Pointers	50
Using Pointers	50
Pointers and the const Keyword	52
Dynamic Memory	53
Strings and Dynamic Memory	56
Returning Strings from Functions	57
Arrays of Strings	58
Example: AddressBook	59
Compile and Run the AddressBook Example	61
Structs	62
Header Files	64
Compile and Run the HeaderFileTest Example	66
Create Files for the Song Struct	66
Final Example	68
 4. Thinking in Objects	 73
Structs and Classes	74
Designing Classes	76
Accessors	77
Inheritance	78
Composition	80
Object Lifetime	81
Built-in Classes	81
 5. Basic Objective-C	 83
NSString Basics	84
Using Methods	84
Nested Method Calls	86
Multi-Input Methods	86
Accessors	86
Dot Syntax	87
Creating Objects	88
Basic Memory Management	88
Using Autorelease Directly	90
Declaring a Class	90
Add Methods	91
Implementing a Class	92
init	94
dealloc	95
Example: PhotoInfo	96

6. More Objective-C	103
More on Memory Management	103
The Life of an Instance Variable	105
Copying Objects	106
Class Name Prefixes	107
Properties	107
Property Options	109
64-Bit Objective-C	111
Enabling 64-Bit	112
Should I Use 64-Bit?	113
All Further Examples Assume 64-Bit	114
Categories	114
Categories for Private Methods	116
Introspection	118
Protocols	120
Dynamic Messaging	124
Using Selectors to Call Methods	125
Forwarding Messages	126
Exceptions	127
Example: DataCollector	129
Some New Classes and Methods	129
Create the Files	130
Build and Run	136
 7. Foundation Value Classes	 137
NSString	138
Ranges and Substrings	140
Using NSString with C Types	141
Comparing Strings for Equality	141
Strings As File Paths	142
Reading and Writing Files with Strings	144
Mutability	148
Advantages of Mutability	149
Advantages of Immutability	150
Core Foundation	150
Memory Management	152
Core Foundation Mutability	153
Toll-Free Bridging	154
Core Foundation Types As Properties	155
Drawbacks of Core Foundation Types	156
Open Source	156
NSNumber	156
CFNumberRef	159

Cocoa Primitive Types	160
NSNumber	161
NSNumberFormatter	163
When to Use Which Number Type	165
NSData	166
NSMutableData	168
NSArray	169
Fast Enumeration	171
Blocks	171
NSMutableArray	173
CFArrayRef	174
NSIndexSet	175
NSDictionary	177
NSMutableDictionary	178
CFDictionaryRef	179
NSSet	181
NSMutableSet	182
NSNumber	183
NSDate	184
NSDateRef	186
8. Basic Controls	189
How to Use This Chapter	189
Windows and Views	190
Targets and Actions	193
Buttons	194
Declaring Action Methods	195
Connecting Actions	198
Menus	203
Responder Chain	206
Pop-up Buttons	211
Sliders	214
Text Fields	216
Outlets	218
Datasources	222
Table View Datasource Methods	226
Implementing Datasource Methods	227
Bindings	230
Key-Value Protocols	234
Bindings for Simple Controls	236
Bindings for Complex Controls	242
Tips for Debugging Bindings	244

9. Designing Applications Using MVC	247
About This Project	248
Window Controllers	249
View Controllers	250
Core Data	253
Create the Project Files	254
Create the Entities	254
Add Attributes and Relationships	255
Update the App Delegate	261
Add the Quartz Framework	264
Create the Window Controller	264
Create the View Controllers	268
Create the Managed Object Classes	280
Create the User Interface	286
Remove the Default Window	287
Create the Main Window Interface	289
Create the Browser Interface	293
Create the Editor View Interface	297
Create the List View Interface	298
Run the Application	301
Preparing for Release	303
 10. Custom Views and Drawing	 307
Basic Geometry	308
Geometry Structs As Strings	310
Geometry Structs As NSValues	311
Cocoa View Coordinates	312
Derived Rects	313
Comparison Functions	314
Basic Drawing	315
NSColor	316
Subclassing NSView	316
Instantiate the View	320
The Graphics Context	323
Bezier Paths	324
Drawing Polygons	324
Drawing Curved Paths	325
Images	329
Loading Image Data	329
Drawing Images in a View	331
Shadows	338
Add a Shadow to StyledImageView	338
Gradients	341

Drawing a Gradient Background	342
Drawing an Image Sheen	344
Refactoring View Code	349
Why You Should Refactor	350
Goals for Refactoring	350
Refactored Header	352
Refactored Implementation	354
Test the Refactored Version	359
Text	359
Fonts	359
Attributed Strings	360
Add a Title to StyledImageView	365
Handling Mouse and Keyboard Events	370
Keyboard Events	371
Mouse Events	372
Add Event Support to StyledImageView	372
11. The Final Word	377
The List	377
Websites	378
Last Thought	380
Index	381

Setup and First Run

I know you want to start writing apps as soon as possible, but there are three things you need to know first:

I don't assume you already know how to program.

You don't need to have existing experience with Objective-C, Cocoa, Xcode, or even C. If you have some familiarity with a computer language (even HTML), it will help you. You do need to be a reasonably proficient Mac user. If you show up with the desire to learn Mac programming, I'll walk you through what you need to know.

The chapters are modular.

If you know C and object-oriented concepts, but not Objective-C specifically, you can skip to Chapter 5 after learning about Xcode in this chapter. If you've already dabbled in Mac or iPhone programming, and know your way around Objective-C and some basic Cocoa topics, you can probably jump to Chapter 7. Otherwise, start right here.

You need a Mac running Snow Leopard.

All of the chapters assume that you're running Mac OS X 10.6 Snow Leopard, which runs only on Intel-based Macs.

All set? Good, let's get Xcode running.

Download and Install Xcode

You can either get Xcode from the Mac OS X install DVD or download it from the official Mac Dev Center website. The version on the website will always be the newest one, but the download may take several hours. If you want to get started right away, you can install from the DVD.