

Learn Cocoa on the Mac



Jack Nutting

Peter Clark

Apress®

Learn Cocoa on the Mac

Copyright © 2013 by Jack Nutting and Peter Clark

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

ISBN-13 (pbk): 978-1-4302-4542-1

ISBN-13 (electronic): 978-1-4302-4543-8

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

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

President and Publisher: Paul Manning

Lead Editor: Steve Anglin

Developmental Editor: Douglas Pundick

Technical Reviewer: Nick Waynik

Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Louise Corrigan,

Morgan Ertel, Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman,

James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke,

Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Tom Welsh

Coordinating Editor: Jill Balzano

Copy Editor: Mary Behr

Compositor: SPi Global

Indexer: SPi Global

Artist: SPi Global

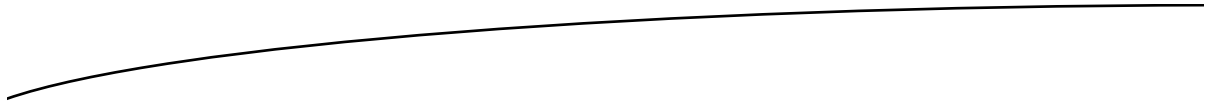
Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media New York, 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. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

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/bulk-sales.

Any source code or other supplementary materials referenced by the author in this text is available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/.



We've spent many nights and weekends staring at computer screens instead of spending time with our wives and children. This book is dedicated to them. Our wives and children that is, not the computer screens.

Contents at a Glance

Preface	xvii
About the Authors.....	xix
About the Technical Reviewer	xxi
Acknowledgments	xxiii
■ Chapter 1: Must Love Cocoa	1
■ Chapter 2: Hello, World	5
■ Chapter 3: Lights, Camera ... Actions! (and Outlets, too).....	27
■ Chapter 4: The First Call to Action.....	47
■ Chapter 5: GUI Components	67
■ Chapter 6: Using Table Views	103
■ Chapter 7: Cocoa Bindings	125
■ Chapter 8: Core Data Basics	153
■ Chapter 9: Core Data Relationships.....	189
■ Chapter 10: Search and Retrieve Core Data with Criteria	209
■ Chapter 11: Windows and Menus and Sheets	225
■ Chapter 12: Document-Based Applications.....	253

■ Chapter 13: Exceptions, Signals, Errors, and Debugging	271
■ Chapter 14: Drawing in Cocoa	295
■ Chapter 15: Advanced Drawing Topics	315
■ Chapter 16: Working with Files	341
■ Chapter 17: Concurrency	357
■ Chapter 18: Future Paths.....	375
Index.....	389

Contents

Preface	xvii
About the Authors.....	xix
About the Technical Reviewer	xxi
Acknowledgments	xxiii
■ Chapter 1: Must Love Cocoa	1
Get a Mac and Download the Tools	2
Download the Source Code	2
Getting Help.....	3
What You Need to Know Before You Begin	3
Are You Ready?.....	3
■ Chapter 2: Hello, World	5
Building “Hello, World”	5
Exploring the Nib File	11
The Library.....	11
Dragging Out a Label	12
Using the Blue Guidelines.....	13
The Inspector.....	14
The Attributes Inspector	16
Change the Label’s Color and Font.....	18

Creating the Application Icon	20
Adding an Icon to our Project	21
Property Lists	22
Running the Application	23
Sharing Our Creation With the World	24
Goodbye, Hello World	25
■ Chapter 3: Lights, Camera ... Actions! (and Outlets, too).....	27
Frameworks, Frameworks Everywhere.....	27
The Foundation Framework.....	28
The AppKit Framework	28
The Cocoa Way: Model–View–Controller	29
Outlets, Actions, and Controllers	30
Outlets	30
Actions.....	30
Outlets and Actions in Action.....	31
Wrap Up.....	45
■ Chapter 4: The First Call to Action.....	47
Declaring Outlets and Actions	47
Declaring Outlets	47
Declaring Actions.....	48
Outlets and Actions in Action, Act 2.....	49
Placeholder Objects.....	51
Setting Up The Window.....	52
Designing the Window’s Interface	55
Creating the Controller Class.....	59
Implementing the Action Method.....	61
The Application Delegate.....	62
Configuring the Application to Quit on Window Close.....	63
Using the Documentation Browser.....	64
Wrap Up.....	65

■ Chapter 5: GUI Components	67
Creating the VillainTracker Application	69
Building the Interface	70
Wiring Up the VillainTrackerAppDelegate Class	84
Getting Started with Coding	92
Standardizing Key Names.....	92
Creating the Default Villain	93
Paying Attention to Detail	94
Responding to Input.....	99
Wrap Up.....	101
■ Chapter 6: Using Table Views	103
Preparing VillainTrackerAppDelegate for Multiple Villains.....	104
Making Way for the Table View.....	105
A Nasty Resize Surprise, or, Constraints To The Rescue!.....	110
Creating and Editing Constraints	111
Making Way for the Table View: Code Edition.....	116
The Table View Needs Help.....	117
Adding More Villains	118
Selecting A Villain	119
Stop Your Evil Ways!	120
Editing in the Table View.....	122
Wrap Up.....	123
■ Chapter 7: Cocoa Bindings	125
Binding to Simple Controls.....	126
Creating the DungeonThing Project and Preferences Window	126
Binding to NSUserDefaultsController.....	130
Creating the Main Window.....	133
Setting Up the DungeonThingAppDelegate.....	135
Binding to a Table View	141
Making the Code Bindings-Ready	141
Configuring the Table Views and Text Views.....	143

Creating and Configuring an Array Controller	145
Binding the Table Display via the Array Controller	147
Binding a Text Field via the Array Controller's Selection	147
Populating the Array	148
Making Sure it Works	148
Rinse, Repeat, Rinse, Repeat	148
Okay, But How Did That Work?	149
Key-Value Coding	149
Key-Value Observing	150
Cocoa Bindings: How It Works	151
Wrap Up.....	151
■ Chapter 8: Core Data Basics	153
What We've Been Missing	153
Creating MythBase	155
Defining the Model	156
Using Xcode's Model Editor	157
Creating an Entity	157
Creating Attributes.....	158
Attributes for Unsupported Types	160
Designing the GUI	161
Creating the MythBase display.....	161
Displaying Details	163
Setting up Cocoa Bindings	166
Binding with an Array Controller.....	166
Configuring Bindings for the Table View	168
Finishing the Bindings: Save and Search	171
Exploring the Template Code.....	172
The App Delegate Interface	173
The App Delegate Implementation.....	174

Adding Business Logic	181
Making The MythicalPerson Class	182
Validating Single Attributes	183
Validating Multiple Attributes	184
Creating a Custom Attribute	186
Wrap Up.....	188
■ Chapter 9: Core Data Relationships	189
Modeling New Entities and Relationships	190
Model Versioning and Migrations	191
Preparing for Multiple Model Versions.....	191
Adding New Entities	192
Adding Relationships.....	193
Creating a Simple Migration	195
Time to Run	195
Updating the GUI	197
Creating the Band Window	197
Setting Up Cocoa Bindings	198
Putting People in Bands.....	200
Showing the Bands' Members.....	203
Creating a Venue Window	204
Adding a Gig List to the Band Window.....	205
Wrap Up.....	207
■ Chapter 10: Search and Retrieve Core Data with Criteria	209
Creating QuoteMonger	209
Creating the Project and Its Data Model.....	210
The Data Entry Window	211
Entering Some Initial Quotes	214
Creating the Quote Finder Window	215

Limiting Results with NSPredicate	216
Creating Predicates	216
Specifying an NSAppController's Predicate in Xcode	218
User-Defined Predicates	218
Saving a Predicate	222
Wrap Up.....	223
■ Chapter 11: Windows and Menus and Sheets	225
NSWindow and NSPanel.....	225
Handling Input	227
To Use a Panel, or Not to Use a Panel.....	228
Window Attributes	228
Standard System Panels.....	231
A Controller With a Nib of Its Own	236
Loading a Nib With NSWindowController.....	237
Subclassing NSWindowController	239
Modal Windows	242
Open Panels and Save Panels	243
System Menus.....	244
Standard Application Menu Items.....	245
Our Own Menus	245
Enabling/Disabling With Bindings.....	245
Enabling/Disabling With First Responder	248
Sheets	251
Wrap Up.....	252
■ Chapter 12: Document-Based Applications	253
Creating the ColorMix Application	254
Examining the Default Nib Files.....	255
Defining the Model	255
Setting Two Colors.....	255
The Simplest of GUIs	256
Creating a Default ColorSet	257

Settling on a File Format	258
Adding Color	259
The CMColorBlendView Class	259
Adding Blended Colors to the GUI	262
About Undo and Redo	268
The Undo Stack	268
Wrap Up	269
Chapter 13: Exceptions, Signals, Errors, and Debugging	271
Exception Handling	271
Catching Exceptions	272
The Limited Role of Exceptions in Cocoa	273
Create a Testbed	273
NSInvalidArgumentException	279
NSRangeException	282
And the Rest	283
Worse than Exceptions: Death by Signal	284
NSError	288
Domains and Codes	288
Recognizing an Error	289
Presenting an Error	293
Wrap Up	293
Chapter 14: Drawing in Cocoa	295
Fundamentals	295
The View Coordinate System	296
Frame Rectangle vs. Bounds Rectangle	296
Rects, Points, and Sizes	297
Path Basics	297
Creating an NSView Subclass	298
The Basic Drawing Method, drawRect:	298
Graphics States	299
Path Helpers	299

Colors and the Graphics Context	300
Beyond Color	301
Manual Path Construction	302
Pushing Boundaries.....	303
LOLmaker	305
First Steps to LOL	306
LOLView	308
Drawing a Bitmap.....	309
Let It Scroll	310
Drawing Text.....	312
Wrap Up.....	314
■ Chapter 15: Advanced Drawing Topics	315
Editing a Curve	315
Preparations	316
Bezier Plumbing	318
Drawing a Curve	320
Watching the Mouse.....	321
A Little Polish.....	323
Core Animation: A Primer	325
Core Animation Basics.....	325
Implicit Animations	325
Explicit Animations	326
Grouping Animations	332
Wrap Up.....	340
■ Chapter 16: Working with Files	341
Implicit File Access.....	341
High-Level File Operations	342
What About That File: The Code	343
What About That File: The GUI.....	349
Wrap Up.....	356

■ Chapter 17: Concurrency	357
SlowWorker	358
Threading Basics	361
Units of Work	362
Operation Queues	362
Becoming a Blockhead	362
Vitalizing SlowWorker	364
Demanding the Main Thread	365
GCD: Low-Level Queuing	370
Improving SlowWorker a Second Time	370
A Little Concurrency Goes a Long Way	374
■ Chapter 18: Future Paths	375
More Cocoa-isms	375
Notifications	376
Blocks	377
Cocoa in a Foreign Language	382
PyObjC	382
MacRuby	383
Nu	384
JavaScript	384
F-Script	385
Ported Cocoa	385
Cocoa Touch	385
GNUstep and Cocotron	386
Cappuccino/Objective-J	387
Here at the End of All Things	387
Index	389

Preface

I first encountered Cocoa as a college student in about 1989. Of course, that was before the iPhone, before Mac OS X, and before it was even called Cocoa. Back then, the seed of today's Cocoa was a part of NeXTStep, the OS that was the core of the NeXT computers. NeXTStep was years ahead of its time, and while the lab full of NeXT workstations was woefully underused in the computer science courses, my student sysadmin job had me using them daily. As a user, I was hooked. I won't dwell on the NeXT user experience here, but I'll just state that many of the best features of Mac OS X come not so much from the Mac of old as from NeXTStep.

At that time, there was no www, not much of a NeXTStep developer community, and very little written about the development environment apart from the impenetrable tomes that NeXT shipped with its earliest machines. I tried to wrap my head around Objective-C and the AppKit from time to time, but without any nearby experts or much example code to look at (not to mention my actual studies, which sometimes distracted me from playing with fun projects), I was basically stumped.

After college, something completely unexpected happened. A friend pointed me in the direction of a consulting firm in my city that was building custom NeXTStep apps for some pretty big customers, and I had the good fortune to come onboard. Suddenly, I had a group of colleagues who had not only been programming in NeXTStep for a while, some of them had even worked at NeXT! And one of them was Peter Clark. All it took was a bit of their expert help to get me started, and the things that had seemed so mysterious for years suddenly made sense. Within a few weeks I learned so much that I was able to start leading some training and mentoring efforts in NeXTStep development.

The point here isn't that I'm a genius or a quick study. It's that the set of technologies we now call Cocoa is really powerful—and quite easy to learn and put to good use—but you're likely to need some help along the way. I'm hoping that this book will help nudge you in the right direction and help you learn the essence of Cocoa programming, so that by the time you're finished reading it, you'll have enough knowledge of Cocoa to be able to propel yourself forward and write the Mac applications of your dreams. In short, the kind of book I wish I'd been able to find 20 years ago.

— Jack Nutting

About the Authors



Jack Nutting has been using Cocoa since the olden days, long before it was even called Cocoa. He's used Cocoa and its predecessors to develop software for a wide range of industries and applications including gaming, graphic design, online digital distribution, telecommunications, finance, publishing, and travel. When he's not working on Mac or iOS projects, making killer games with Unity3D, or rocking out in his punk rock band, he's usually spending time with his family. Jack is @jacknutting on Twitter and blogs from time to time at www.nuthole.com. (Photograph courtesy of Alison De Mars von Blixen.)



Peter Clark started with NeXTStep in college, circa 1990. That experience set the tone for the first half of his career when he worked with Jack at a consulting company in the Twin Cities. He worked with companies in the Bay Area, Chicago, and Twin Cities on NeXTStep and WebObjects projects, wrote a book on OpenStep development, and then moved on to Java, .NET, Ruby, and Scala development and managing engineering teams. He's pleased that so many of the design patterns from NeXTStep are still useful today and finds that Cocoa development is a mix of comfort and pleasant surprises. Peter is currently the Director of Software Development for a research project at the University of Minnesota. He blogs occasionally at <http://blog.pclark.net> and can be found on Twitter as @pclark. Peter wants you to know that he started using NeXTStep at least a month or two before Jack did. (Photograph courtesy of Wade Stebbings.)

About the Technical Reviewer



Nick Waynik has been working in the IT field for over thirteen years and has done everything from network administration to web development. He started writing iOS apps when the SDK was first released. Since then he has gone on to start his own business, focusing on iOS development. He loves spending his free time with his wife, Allison, and son, Preston, and sometimes playing golf. He blogs at nickwaynik.com and can be found on Twitter as @n_dubbs.

Acknowledgments

There are many people to thank when getting a book from concept to the paper (or e-book) that you're reading. There's a lot to be done besides just getting words and code snippets into a text editor. Douglas Pundick and Jill Balzano at Apress have helped keep many of those things on track, out of sight from the authors. Of course, keeping the authors on track is also important, and we kept them busy at that too. Nick Waynik, our technical reviewer, caught many small (and some not-so-small) details with both the code and the prose, and always had a good reason for suggesting a change. Mary Behr, our copyeditor, did a wonderful job of tightening our prose, polishing our grammar, and making sure that what you're reading hangs together cleanly. In her hands, many an awkward sentence got refactored into something better.

We'd like to thank the developers at Apple for producing tools worth writing about. The writer at www.swampfoetus.net did a great writeup for readers of the first edition about the changes in Xcode 4 dealing with Core Data using the MythicalPerson sample app from the first edition, and I want to thank him for caring enough to write that and make it public. The other authors of the first edition, David Mark and Jeff LaMarche, left us with a solid foundation upon which to build. Thanks, gentlemen! We're sure we're missing plenty of people who richly deserve to be thanked for making this book good. The blame for the remaining bad parts rests solely with the authors.

Much of the writing was done at Shish Café and at Kopplin's Coffee in Saint Paul, Minnesota. If you want good tasting coffee, you won't find better than at Kopplin's. Lastly, and most importantly, Peter wants to thank his wife, Molly McBeath, and kids Laurel and Béa, and Jack wants to thank his wife, Weronica Meijer, and kids Henrietta and Dorotea. They've been very supportive of the time commitment it takes to write a book and it wouldn't have happened without their help. Peter and Jack dedicate this book to our families.