

ROB NAPIER | MUGUNTH KUMAR

iOS 6 PROGRAMMING

 PUSHING THE LIMITS

Developing Extraordinary Mobile Apps
for Apple iPhone®, iPad®, and iPod Touch®

```
*Begin PBXFrameworksBuildPhase section */
7213FCFE15FF5E6E004A58FF /* Frameworks */ = {
isa = PBXFrameworksBuildPhase;
buildActionMask = 2147483647;
files = (
7213FD0615FF5E6E004A58FF /* UIKit.framework in Frameworks */,
7213FD0815FF5E6E004A58FF /* Foundation.framework in Frameworks */,
7213FD0A15FF5E6E004A58FF /* CoreGraphics.framework in Frameworks */
);
runOnlyForDeploymentPostprocessing = 0;
};
*End PBXFrameworksBuildPhase section */
```



iOS 6 Programming

Pushing the Limits

iOS 6 Programming **Pushing the Limits**

ADVANCED APPLICATION DEVELOPMENT
FOR APPLE IPHONE®, IPAD®, AND IPOD
TOUCH®

Rob Napier and Mugunth Kumar



A John Wiley and Sons, Ltd, Publication

This edition first published 2013

© 2013 Rob Napier and Mugunth Kumar

Registered office

John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

The right of the author to be identified as the author of this work has been asserted in accordance with the Copyright, Designs and Patents Act 1988.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as permitted by the UK Copyright, Designs and Patents Act 1988, without the prior permission of the publisher.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

DESIGNATIONS USED BY COMPANIES TO DISTINGUISH THEIR PRODUCTS ARE OFTEN CLAIMED AS TRADEMARKS. ALL BRAND NAMES AND PRODUCT NAMES USED IN THIS BOOK ARE TRADE NAMES, SERVICE MARKS, TRADEMARKS OR REGISTERED TRADEMARKS OF THEIR RESPECTIVE OWNERS. THE PUBLISHER IS NOT ASSOCIATED WITH ANY PRODUCT OR VENDOR MENTIONED IN THIS BOOK. THIS PUBLICATION IS DESIGNED TO PROVIDE ACCURATE AND AUTHORITATIVE INFORMATION IN REGARD TO THE SUBJECT MATTER COVERED. IT IS SOLD ON THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING PROFESSIONAL SERVICES. IF PROFESSIONAL ADVICE OR OTHER EXPERT ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL SHOULD BE SOUGHT.

Trademarks: Wiley and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. Ltd. and/ or its affiliates in the United States and/or other countries, and may not be used without written permission. iPhone, iPad and iPod touch are trademarks of Apple Inc. All trademarks are the property of their respective owners. John Wiley & Sons, Ltd. is not associated with any product or vendor mentioned in the book.

ISBN 978-1-118-44995-0 (pbk); ISBN 978-1-118-44996-7 (ebk); ISBN 978-1-118-44997-4 (ebk); ISBN 978-1-118-44998-1 (ebk)

A catalogue record for this book is available from the British Library.

Set in 9.5pt Myriad Pro Regular by Wiley Publishing

Printed in United States by Bind-Rite

Publisher's Acknowledgements

Some of the people who helped bring this book to market include the following:

Editorial and Production

VP Consumer and Technology Publishing Director: Michelle Leete

Associate Director–Book Content Management: Martin Tribe

Associate Publisher: Chris Webb

Acquisitions Editor: Craig Smith

Assistant Editor: Ellie Scott

Development Editor: Tom Dinse

Copy Editor: Melba Hopper

Technical Editor: Jay Thrash

Editorial Manager: Jodi Jensen

Senior Project Editor: Sara Shlaer

Editorial Assistant: Leslie Saxman

Marketing

Associate Marketing Director: Louise Breinholt

Marketing Manager: Lorna Mein

Senior Marketing Executive: Kate Parrett

Composition Services

Compositor: Indianapolis Composition Services

Proofreader: Cynthia Fields

Indexer: Potomac Indexing, LLC

About the Authors

Rob Napier is a builder of tree houses, hiker, and proud father. He began developing for the Mac in 2005, and picked up iPhone development when the first SDK was released, working on products such as The Daily, PandoraBoy, and Cisco Mobile. He is a major contributor to Stack Overflow and maintains the *Cocoaphony* blog (cocoaphony.com).

Mugunth Kumar is an independent iOS developer based in Singapore. He graduated in 2009 and holds a Masters degree from Nanyang Technological University, Singapore, majoring in Information Systems. He writes about mobile development, software usability, and iOS-related tutorials on his blog (blog.mugunthkumar.com). Prior to iOS development he worked for Fortune 500 companies GE and Honeywell as a software consultant on Windows and .NET platforms. His core areas of interest include programming methodologies (Object Oriented and Functional), mobile development and usability engineering. If he were not coding, he would probably be found at some exotic place capturing scenic photos of Mother Nature.

Dedication

To Neverwood. Thanks for your patience.

Rob

To my mother who shaped the first twenty years of my life

Mugunth

About the Technical Editor

Jay Thrash is a veteran software developer who has spent the past three years designing and developing iOS applications. During his career, he developed a keen interest in the areas of user interaction and interface design.

Prior to settling down as an iOS developer, Jay has worked on a variety of platforms and applications, including flight simulators and web application development. He has also spent over six years in the PC and console gaming industry.

Authors' Acknowledgments

After spending much of last year writing for iOS 5, Rob thanks his family for their patience as he did it all again for iOS 6. Mugunth thanks his parents and friends for their support while writing this book. Thanks to Wiley, especially Craig Smith and Ellie Scott, for the continued support, encouragement, and nudging that it takes to get a book out the door. Thanks to Jay Thrash for jumping into the fray and tech editing for us, and to Tom Dinse who made sure that it was intelligible. Thanks to the Apple engineers at WWDC who put up with a deluge of questions, probing, and occasional complaining. And special thanks to Steve Jobs for a lifetime of elegant boxes. He will be missed.

Contents

Introduction	1
Who This Book Is For	1
What This Book Covers	1
What Is New in This Edition	1
How This Book Is Structured	2
What You Need to Use This Book	5
Finding Apple Documentation	5
Source Code	5
Errata	6
Part I: What's New?	7
Chapter 1 The Brand New Stuff	9
What's New	9
Summary	13
Further Reading	14
Chapter 2 Getting Comfortable with Xcode 4 and the LLVM Compiler	15
Getting to Know the User Interface	16
All in One Window	18
Navigating the Navigators	19
Help from Your Assistant	21
Integrated Interface Builder	21
LLVM Compiler: A Tryst with the Brain	23
What's New in LLVM 4	23
Git Your Versions Here	27
Schemes	29
Build Configurations You Can Comment	31
Migrating Your Code to Objective-C ARC	32
Xcode 4 Organizer	34
Summary	35
Further Reading	35

Part II: Getting the Most Out of Everyday Tools 37

Chapter 3 Everyday Objective-C	39
Naming Conventions	39
Automatic Reference Counting	41
Properties	44
Accessors	47
Categories and Extensions	48
Formal and Informal Protocols	52
Summary	54
Further Reading	54
Chapter 4 Hold On Loosely: Cocoa Design Patterns	55
Understanding Model-View-Controller	55
Understanding Delegates and Data Sources	58
Working with the Command Pattern	59
Working with the Observer Pattern	67
Working with the Singleton Pattern	70
Summary	72
Further Reading	73
Chapter 5 Memory Management with Objective-C ARC	75
Introduction to Objective-C ARC	75
What Is ARC?	76
Common ARC Migration Errors	83
Summary	86
Further Reading	86
Chapter 6 Getting Table Views Right	89
UITableView Class Hierarchy	89
Understanding Table Views	90
Summary	102
Further Reading	103
Chapter 7 Great at Any Angle: Collection Views and Auto Layout	105
Collection Views	105
Cocoa Auto Layout	116
Summary	123
Further Reading	123

Chapter 8 Better Drawing.....	125
iOS's Many Drawing Systems	125
UIKit and the View Drawing Cycle.....	126
View Drawing Versus View Layout	128
Custom View Drawing.....	129
Optimizing UIView Drawing	144
CGLayer	148
Summary	150
Further Reading.....	150
Chapter 9 Layers Like an Onion: Core Animation.....	151
View Animations.....	151
Managing User Interaction	153
Drawing with Layers	154
Moving Things Around.....	158
Into the Third Dimension.....	164
Decorating Your Layers.....	168
Auto-Animate with Actions.....	170
Animating Custom Properties	171
Core Animation and Threads	173
Summary	173
Further Reading.....	173
Chapter 10 Tackling Those Pesky Errors	175
Error-Handling Patterns	175
Assertions	176
Exceptions.....	178
Catching and Reporting Crashes.....	179
Errors and NSError	180
Error Handler Blocks.....	182
Logs	183
Summary	185
Further Reading.....	186
Chapter 11 Location Services: Know Where You Are	187
Core Location Framework	187
Background Location	191
Location Simulation.....	194
Keeping an Eye on the Battery.....	195
Summary	195
Further Reading.....	195

Part III: The Right Tool for the Job	197
Chapter 12 Common UI Paradigms Using Table Views	199
Pull-To-Refresh	199
Infinite Scrolling	202
Inline Editing and Keyboard	204
Animating a UITableView	206
Practical Implementations of Table View Animations	208
Summary	211
Further Reading	212
Chapter 13 Controlling Multitasking	213
Best Practices for Backgrounding: With Great Power Comes Great Responsibility	213
When We Left Our Heroes: State Restoration	215
Introduction to Multitasking and Run Loops	223
Developing Operation-Centric Multitasking	224
Multitasking with Grand Central Dispatch	229
Summary	235
Further Reading	236
Chapter 14 REST for the Weary	237
The REST Philosophy	238
Choosing Your Data Exchange Format	238
A Hypothetical Web Service	241
Important Reminders	242
RESTfulEngine Architecture (iHotelApp Sample Code)	243
Summary	258
Further Reading	258
Chapter 15 Batten the Hatches with Security Services	261
Understanding the iOS Sandbox	261
Securing Network Communications	263
Employing File Protection	270
Using Keychains	272
Using Encryption	274
Summary	287
Further Reading	287

Chapter 16 Running on Multiple iPlatforms and iDevices	289
Developing for Multiple Platforms	290
Detecting Device Capabilities	294
In App Email and SMS	300
Checking Multitasking Awareness	300
Supporting the iPhone 5	301
UIRequiredDeviceCapabilities	303
Summary	304
Further Reading	304
Chapter 17 Internationalization and Localization	305
What Is Localization?	305
Localizing Strings	306
Auditing for Nonlocalized Strings	307
Formatting Numbers and Dates	309
Nib Files and Base Internationalization	312
Localizing Complex Strings	312
Summary	315
Further Reading	315
Chapter 18 Selling Past the Sale with In App Purchases	317
Before You Start	317
In App Purchase Products	317
Setting Up Products on iTunes Connect	320
In App Purchase Implementation	326
Introduction to MKStoreKit	327
Making the Purchase	331
Downloading Hosted Content	332
Testing Your Code	332
Troubleshooting	333
Summary	334
Further Reading	334
Chapter 19 Debugging	335
LLDB	335
Debugging with LLDB	335
Breakpoints	338
Watchpoints	343

The LLDB Console	344
NSZombieEnabled Flag	348
Different Types of Crashes	348
Collecting Crash Reports	352
Third-Party Crash Reporting Services	353
Summary	354
Further Reading	354
Chapter 20 Performance Tuning Until It Flies	357
The Performance Mindset	357
Welcome to Instruments	358
Finding Memory Problems	360
Finding CPU Problems	364
Drawing Performance	368
Optimizing Disk and Network Access	371
Summary	371
Further Reading	372
Part IV: Pushing the Limits	373
Chapter 21 Storyboards and Custom Transitions	375
Getting Started with Storyboards	375
Custom Transitions	380
Customizing Your Views Using UIAppearance Protocol	382
Summary	383
Further Reading	383
Chapter 22 Cocoa’s Biggest Trick: Key-Value Coding and Observing	385
Key-Value Coding	385
Key-Value Observing	395
KVO Tradeoffs	399
Summary	400
Further Reading	401
Chapter 23 Think Different: Blocks and Functional Programming	403
What Is a Block?	403
Declaring a Block	406
Implementing a Block	408
Blocks and Concurrency	410

Block-Based Cocoa Methods.....	412
Supported Platforms	415
Summary	415
Further Reading.....	415
Chapter 24 Going Offline	417
Reasons for Going Offline	417
Strategies for Caching.....	418
Data Model Cache.....	422
Cache Versioning	426
Creating an In-Memory Cache	427
Creating a URL Cache	431
Summary	433
Further Reading.....	433
Chapter 25 Data in the Cloud	435
iCloud	435
Third-Party Cloud Offerings.....	437
Parse	438
StackMob.....	442
Disadvantages of Using a Backend as a Service.....	444
Summary	444
Further Reading.....	444
Chapter 26 Fancy Text Layout	447
The Normal Stuff: Fields, Views, and Labels.....	447
Rich Text in UIKit	448
Web Views for Rich Text	453
Core Text	454
Summary	464
Further Reading.....	464
Chapter 27 Building a (Core) Foundation.....	465
Core Foundation Types.....	465
Naming and Memory Management.....	466
Allocators.....	467
Introspection	468
Strings and Data	469
Collections.....	474

Toll-Free Bridging	477
Summary	480
Further Reading	480
Chapter 28 Deep Objective-C	481
Understanding Classes and Objects	481
Working with Methods and Properties	483
How Message Passing Really Works	486
Method Swizzling	494
ISA Swizzling	496
Method Swizzling Versus ISA Swizzling	498
Summary	498
Further Reading	498
Index	501

Introduction

Apple has a history of alternating its releases between being user-focused and developer-focused. iOS 6 is mostly about users. There's a totally new map interface with turn-by-turn driving directions. Social networks permeate the interface. Pass Kit opens up whole new ways to integrate the iPhone with daily life. The Siri enhancements make the iPhone 4S even more compelling. Expect users to upgrade quickly.

That's great news, because iOS 6 is an exciting release for developers, too. So many things have gotten so much easier. It's easier to lay out your UI with collection views and auto layout. UIKit has added rich text support. When your apps go into the background, state preservation makes it easy to keep track of where you were. Many things that were hard or tedious just became a lot simpler.

If you're ready to take on the newest Apple release and push your application to the limits, this book will get you there.

Who This Book Is For

This is not an introductory book. Many other books out there cover Objective-C and will walk you step by step through Interface Builder. However, this book assumes that you have a little experience with iOS. Maybe you're self-taught, or maybe you've taken a class. You've hopefully written at least most of an application, even if you haven't submitted it yet. If you're ready to move beyond the basics, to learn the best practices and the secrets that the authors have gleaned from practical experience writing real applications, then this is the book for you.

This book also is not just a list of recipes. There's plenty of sample code here, but the focus is on discovering how to design, code, and maintain great iOS apps. Much of this book is about *why* rather than just *how*. You'll find out as much about design patterns and writing reusable code as about syntax and new frameworks.

What This Book Covers

The iOS platforms always move forward, and so does this book. Most of the examples here require at least iOS 5, and many require iOS 6. All examples use Automatic Reference Counting, automatic property synthesis, and object literals. Except in a very few places, this book will not cover backward compatibility. If you've been shipping code long enough to need backward compatibility, you probably know how to deal with it. This book is about writing the best possible apps using the best features available.

This book focuses on iPhone 4, iPad 2, and higher models. Most topics here are applicable to the original iPad, iPod touch, iPhone 3GS, and Apple TV. Chapter 16 is devoted to dealing with the differences between the platforms.

What Is New in This Edition

This edition covers most of the newest additions to iOS 6, including collection views and Auto Layout (Chapter 7), and a complete chapter is dedicated to Automatic Reference Counting. We have added a whole new chapter

explaining how to move your data to the cloud, covering some third-party backend as a service (BaaS) offerings. More relevant to iOS, this year saw the greatest improvement in the LLVM compiler and debugger. We expanded the LLVM section in Chapter 2 to highlight the important features of LLVM that you could use to write the next-generation apps. We also added a whole new chapter dedicated to debugging with LLDB (Chapter 19), covering advanced topics, including Python scripting, knowing the layout of registers, and reading the registers to solve the most difficult debugging problems. That chapter also provides a brief look at some command-line tools such as `otool` and `atos` that can help you with debugging. Lastly, Chapter 20 is dedicated to showing how to optimize your CPU performance, memory usage, network bandwidth, and battery-using Instruments.

Even if you have the previous edition of this book, the new coverage on debugging (Chapter 19), performance tuning (Chapter 20), and collection views and Auto Layout (Chapter 7) will probably be enough to convince you that you need a copy of this edition.

How This Book Is Structured

iOS has a very rich set of tools, from high-level frameworks like UIKit to very low-level tools like Core Text. Often, there are several ways to achieve a goal. As a developer, how do you pick the right tool for the job?

This book separates the everyday from the special purpose, helping you pick the right solution to each problem. You'll discover why each framework exists, how the frameworks relate to each other, and when to choose one over another. Then you'll learn how to make the most of each framework for solving its type of problem.

There are four parts to this book, moving from the most common tools to the most powerful. Chapters that are new in this edition or have been extensively updated are indicated.

Part I: What's New?

If you're familiar with iOS 5, then this section quickly introduces you to the new features of iOS 6.

- **(Updated) Chapter 1: "The Brand New Stuff"**—iOS 6 adds a lot of new features, and here you get a quick overview of what's available.
- **(Updated) Chapter 2: "Getting Comfortable with Xcode 4 and the LLVM compiler"**—Xcode 4 is a huge IDE, and many of its features aren't obvious. Whether you're new to Xcode or a veteran from earlier versions, this chapter will help you make the most of this powerful (and sometimes frustrating) tool.

Part II: Getting the Most Out of Everyday Tools

As an iOS developer, you've encountered a wide variety of common tools, from notifications to table views to animation layers. But are you using these tools to their full potential? In this part, you find the best practices from seasoned developers in Cocoa development.

- **Chapter 3: "Everyday Objective-C"**—If you're ready to move to the next level in Objective-C, this chapter introduces you to the tools experienced developers use every day to improve application design, maintainability, and reusability.
- **Chapter 4: "Hold On Loosely: Cocoa Design Patterns"**—Cocoa relies on a number of common and consistent design patterns. You discover what they are so you can solve problems the same way Apple does.

- **(New) Chapter 5: “Memory Management with Objective-C ARC”**—Automatic Reference Counting has radically changed how iOS developers work. While most of the conversion is easy, migrating your code and your practices can create some subtle problems. You find out how to avoid the problems and get the most out of this powerful technology.
- **Chapter 6: “Getting Table Views Right”**—Table views are perhaps the most complex and commonly used UI element in iOS. They are simple and elegant in design, but confusing to developers who don’t understand how they work. You learn how to use them correctly and how to solve some special problems like infinite scrolling.
- **(New) Chapter 7: “Great at Any Angle: Collection Views and Auto Layout”**—Nothing created as much buzz at WWDC 2012 as `UICollectionView`. Up until now, `UITableView` has dominated iOS interfaces, but that will likely change in iOS 6 as developers master this new control. With the addition of constraint-based layout from OS X, incredibly complex and beautiful interfaces are now much simpler to implement. This chapter shows you the ropes.
- **Chapter 8: “Better Drawing”**—Custom drawing is intimidating to many new developers, but it’s a key part of building beautiful and fast user interfaces. You’ll discover the available drawing options from UIKit to Core Graphics, and how to optimize them to look their best while keeping them fast.
- **Chapter 9: “Layers Like an Onion: Core Animation”**—iOS devices have incredible facilities for animation. With a powerful GPU and the highly optimized Core Animation, you can build engaging, exciting, and intuitive interfaces. In this chapter, you go beyond the basics and discover the secrets of animation.
- **Chapter 10: “Tackling Those Pesky Errors”**—You try to write perfect code, but sometimes things go wrong. How your application reacts to the unexpected is what separates decent apps from extraordinary apps. You’ll find the common patterns for error handling, how to log, and how to make your code more resilient against the unexpected.
- **(New) Chapter 11: “Location Services: Know Where You Are”**—Many applications need to know where the user is. While iOS makes this easy, it is important not to drain the battery in the process. You find out how to balance functionality with efficiency and give the user the best possible experience.

Part III: The Right Tool for the Job

There are tools that are part of nearly every application, and there are tools that you only need from time to time. In this section, you learn about the tools and techniques that are a little more specialized.

- **(New) Chapter 12: “Common UI Paradigms Using Table Views”**—Exciting as collection views are, the table view is still the workhorse of iOS. You find out how to take table views beyond the basics with infinite scrolling views and other advanced topics.
- **(Updated) Chapter 13: “Controlling Multitasking”**—Multitasking is an important part of many applications, and you discover how to do multiple things at once while your application is running and when your application is in the background. The new state preservation system of iOS 6 makes this much easier, and you learn how to use it for common situations.
- **(Updated) Chapter 14: “REST for the Weary”**—REST-based services are a mainstay of modern applications, and you learn how to best implement them in iOS.

- **(Updated) Chapter 15: “Batten the Hatches with Security Services”**—User security and privacy are paramount today, and you find out how to protect your application and user data from attackers with the keychain, certificates, and encryption.
- **Chapter 16: “Running on Multiple iPlatforms and iDevices”**—The iOS landscape gets more complex every year with iPod touch, iPhone, iPad, Apple TV, and a steady stream of new editions. It’s not enough just to write once, run everywhere. You need your applications to be their *best* everywhere. You learn how to adapt your apps to the hardware and get the most out of every platform.
- **Chapter 17: “Internationalization and Localization”**—Although you may want to focus on a single market today, you can do small things to ease the transition to a global market tomorrow. Save money and headaches later, without interrupting today’s development.
- **Chapter 18: “Selling Past the Sale with In App Purchases”**—In App Purchases are still an untapped market for many developers. Users like the add-on content, and developers love the extra revenue. You discover the best ways to make this important feature a reality in your application.
- **(New) Chapter 19: “Debugging”**—If only every application were perfect the first time. Luckily, Xcode and LLDB provide many tools to help you track down even the trickiest of bugs. You go beyond the basics and find out how to deal with errors in development and in the field.
- **(New) Chapter 20: “Performance Tuning Until It Flies”**—Performance separates the “okay” app from the exceptional app. It’s critical to optimizing CPU and memory performance, but you also need to optimize battery and network usage. Apple provides an incredible tool for this in Instruments. You discover how to use Instruments to find the bottlenecks, and then how to improve performance once you find the problems.

Part IV: Pushing the Limits

This section is what this book is all about. You’ve learned the basics. You’ve learned the everyday. Now push the limits with the most advanced tools available. You discover the ins and outs of deep iOS.

- **(New) Chapter 21: “Storyboards and Custom Transitions”**—Storyboards can still be confusing and a bit intimidating for developers familiar with nib files. You learn how to use storyboards to your advantage and how to push them beyond the basics.
- **Chapter 22: “Cocoa’s Biggest Trick: Key-Value Observing”**—Many of Apple’s most powerful frameworks rely on KVO for their performance and flexibility. You find out how to leverage the flexibility and speed of KVO, as well as the trick that makes it so transparent.
- **Chapter 23: “Think Different: Blocks and Functional Programming”**—Many developers are still absorbing the addition of blocks to Objective-C. They’re valuable for interacting with Apple frameworks, but they also open new ways of thinking about your program. Embrace a new style, and maximize its benefits in your next project.
- **Chapter 24: “Going Offline”**—Network programming is difficult, but even harder is providing a seamless offline experience. Discover how to best cache your data and integrate it into your network engine.
- **(New) Chapter 25: “Data in the Cloud”**—Developers are still adapting the new iCloud service. Supporting it quickly can move your app ahead of the pack. You learn your way around this powerful system.
- **(Updated) Chapter 26: “Fancy Text Layout”**—From UIKit to Core Text, iOS is full of ways to display text. iOS adds rich text support throughout UIKit. You find out how to use attributed strings, web views, and Core Text to handle the toughest layout problems.

- **Chapter 27: “Building a (Core) Foundation”**—When you want the most powerful frameworks available on iOS, you’re going to want the Core frameworks like Core Graphics, Core Animation, and Core Text. All of these rely on Core Foundation. In this chapter, you discover how to work Core Foundation data types so you can leverage everything iOS has to offer.
- **Chapter 28: “Deep Objective-C”**—When you’re ready to pull back the curtain on how Objective-C really works, this is the chapter for you. You find out how to use the Objective-C runtime directly to dynamically modify classes and methods. You also learn how Objective-C method calls are dispatched to C function calls, and how you can take control of the system to extend your programs in incredible ways.

You can skip around in this book to focus on the topics you need most. Each chapter stands alone, except for those that require Core Foundation data objects (particularly Core Graphics, Core Animation, and Core Text). Those chapters direct you to Chapter 27, “Building a (Core) Foundation,” when you need that information.

What You Need to Use This Book

All examples in this book were developed with Xcode 4.5 on Mac OS X 10.8 and iOS 6. You need an Apple developer account to access most of the tools and documentation, and you need a developer license to run applications on your iOS device. Visit developer.apple.com/programs/ios to sign up.

Most of the examples in this book will run in the iOS Simulator that comes with Xcode 4.5. You can use the iOS Simulator without an Apple developer license.

Finding Apple Documentation

Apple provides extensive documentation at its website and within Xcode. The URLs change frequently and are often very long. This book refers to Apple documents by title rather than by URL. To find documents in Xcode, press `Cmd-Option-?` or click `Help` → `Documentation and API Reference`. In the Documentation Organizer, click the Search icon, type the name of the document, and then select the document from the search results. See Figure 1 for an example of how to search for the *Coding Guidelines for Cocoa*.

To find documents at the Apple developer site, visit developer.apple.com, click Member Center, and log in. Select the iOS Dev Center, and enter the document title in the Search Developer search box.

The online documentation is generally identical to the Xcode documentation. You may receive results for both iOS and Mac. Be sure to choose the iOS version. Many iOS documents are copies of their Mac counterparts and occasionally include function calls or constants that are not available on iOS. This book tells you which features are available on iOS.

Source Code

As you work through the examples in this book, you may choose either to type in all the code manually or to use the source code files that accompany the book. All of the source code used in this book is available for download at either <http://iosptl.com/code> or www.wiley.com/go/ptl/ios6programming. For example, you will find the following sample code online in the Chapter 26 folder, in the `SimpleLayout` project, and the `CoreTextLabel.m` file.

CoreTextLabel.m (SimpleLayout)

```

- (id)initWithFrame:(CGRect)frame {
    if ((self = [super initWithFrame:frame])) {
        CGAffineTransform
        transform = CGAffineTransformMakeScale(1, -1);
        CGAffineTransformTranslate(transform,
                                0, -self.bounds.size.height);
        self.transform = transform;
        self.backgroundColor = [UIColor whiteColor];
    }
    return self;
}

```

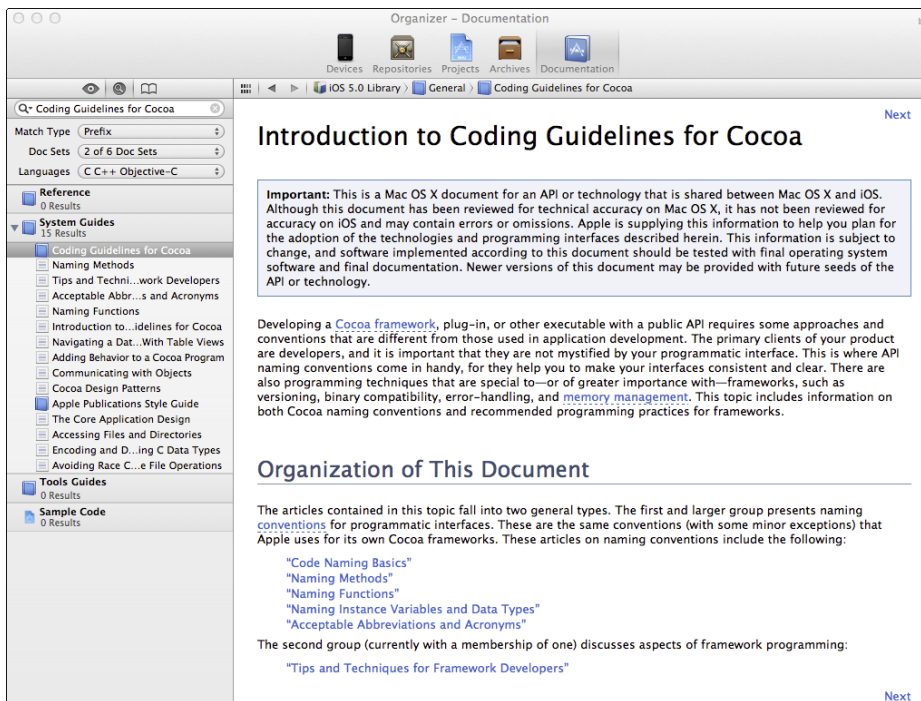


Figure 1 Searching for Coding Guidelines for Cocoa

Some source code snippets shown in the book are not comprehensive and are meant to help you understand the chapter. For those instances, you can refer to the files available on the website for the complete source code.

Errata

We try to get everything right, but sometimes things change, sometimes we mistype, and sometimes we're just mistaken. See <http://iospt1.com/code> for the latest updates, as well as blog posts on topics that haven't made the book yet. Send any mistakes you find to robnapier@gmail.com and contact@mk.sg.

Part I

What's New?

Chapter 1 The Brand New Stuff

Chapter 2 Getting Comfortable with Xcode 4 and the
LLVM Compiler

Chapter 1

The Brand New Stuff

In 2007, Steve Jobs went on stage at Macworld and proclaimed that software running on the iPhone was at least five years ahead of the competition. Since its initial release up to the time of this writing, Apple has been iterating the operating system year after year and has even added two new devices, the iPad and the Apple TV, to the list of products capable of running iOS. Over time, as the operating system was customized to run on more devices than the iPhone, it was rebranded as iOS. Now, after five years and true to the words of Steve Jobs, iOS is still the most advanced operating system around. iOS 6 takes the most advanced operating system to the next level, adding great features, including some developer-centric features like collection views, Auto Layout, and improved data privacy.

This book is about programming with iOS 6, and in this chapter, I go through the main features of iOS 6 and refer you to the chapters covering them in detail.

What's New

Every version of iOS has introduced several key features and other minor features. The second version, iPhone OS 2, was the first to have a public SDK.

iPhone OS 3 brought Core Data from the Mac to the iPhone; other additions included Push Notifications, `ExternalAccessoryKit`, In App Purchases through `StoreKit.framework`, in app e-mail sheets, `MapKit.framework` that allows developers to embed Google Maps into their apps, read-only access to the iPod library, and keychain data sharing.

A minor update, 3.1, added video editor support, and another minor (yet so major) update, 3.2, added Core Text and gesture recognizers, file sharing, and PDF generation support. The 3.2 update also added a whole new product, iPad, and support for developing apps that run on iPad, as well as Universal apps that run on iPad (3.2) and iPhone (3.1.3). However, 3.2 was iPad-only and didn't run on iPhone or iPod touch devices.

iPhone OS 4 (rebranded as iOS 4) introduced the much-awaited multitasking support, local push notifications, read-only access to calendar (`EventKit.framework`), blocks and Grand Central Dispatch, In app SMS sheets, and Retina display support. This version was iPhone-only and didn't support developing apps for iPad. A minor update to this, iOS 4.2, unified iPhone and iPad operating systems.

iOS 5 introduced several important features like iCloud, Automatic Reference Counting, built-in Twitter framework, and several other minor features.

iOS 6 introduces Pass Kit, a framework which providers can use to push tickets, passes, or coupons directly to the iPhone. For example, a hotel reservation iOS app can send a confirmation of registration as a pass to the user's iPhone.

iOS 6 also introduces a brand-new maps application for end users. To developers, there is a new SDK that allows you to show directions. You open maps by calling the method `openMapsWithOptions: onMKMapItem:`

```
+ (BOOL)openMapsWithOptions:(NSArray *)mapItems launchOptions:
    (NSDictionary *)launchOptions
```

However, iOS 6 doesn't support opening maps through a URL scheme. So, you need to fall back to Google for iOS 5 devices, as illustrated in the following code fragment:

```
if ([[UIApplication sharedApplication] canOpenURL:
    [NSURL URLWithString:@"maps"]]) {
    // ios 6 specific
    [MKMapItem openMapsWithOptions:<items> launchOptions:<options>]; } else
{
    // ios 5 specific, fall back to Google
    [[UIApplication sharedApplication] openURL:
    [NSURL URLWithString:@"http://maps.google.com/<your url>"]]
}
```

This book focuses primarily on the new features added in iOS 6. In the next few sections, I introduce you to the key features added to iOS 6 and provide guidance for using them to push your apps to the next level.

Collection Views

Arguably the most important addition to iOS 6 is a brand-new controller for displaying and managing collection views. Collection views finally provide a way to lay out your UI elements as a grid view like the built-in Photos app or the iBooks app. You no longer have to depend on third-party grid view frameworks for creating grids or mess around with scroll views. Collection views also provide an easy way to animate content (similar to `UITableView`'s row animation) and to insert, delete, and reorder them. (See Chapter 7 for more on collection views.)

Auto Layout

Auto Layout is a new way to define rules that determine how elements in your user interface should be laid out. This capability was introduced in the Mac OS X SDK last year and brought to iOS with iOS 6. Auto Layout is more intuitive than the previously used "springs and struts" model and provides built-in support for mirroring UI elements for right-to-left languages. Arguably, the most important feature that Auto Layout brings in is the capability to swap strings when you localize your nib files.