



Jiva DeVoe

# Cocoa Touch™ for iPhone® OS 3

 Developer Reference



**Cocoa Touch™  
for iPhone® OS 3**



# **Cocoa Touch™ for iPhone® OS 3**

**Jiva DeVoe**



WILEY

Wiley Publishing, Inc.

### **Cocoa Touch™ for iPhone® OS 3**

Published by  
**Wiley Publishing, Inc.**  
10475 Crosspoint Boulevard  
Indianapolis, IN 46256  
www.wiley.com

Copyright © 2010 by Wiley Publishing, Inc., Indianapolis, Indiana

Published by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-0-470-48107-3

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

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, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, 201-748-6011, fax 201-748-6008, or online at <http://www.wiley.com/go/permissions>.

---

**LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.**

---

For general information on our other products and services or to obtain technical support, please contact our Customer Care Department within the U.S. at (877) 762-2974, outside the U.S. at (317) 572-3993 or fax (317) 572-4002.

Library of Congress Control Number: 2009937274

Trademarks: Wiley and the Wiley logo are registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Cocoa Touch and iPhone are trademarks or registered trademarks of Apple, Inc. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book. Cocoa Touch for iPhone OS 3 is an independent publication and has not been authorized, sponsored, or otherwise approved by Apple, Inc.

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

***For my wife, Dawn, and my children, Robert, Alex, and Izzy.  
You are, and always will be, the most important  
people in the world to me.***

## About the Author

**Jiva DeVoe** has been writing software for nearly 25 years, starting with his Commodore VIC-20 using BASIC and assembly language, and gradually working his way through C, C++, Python, Ruby, Java, and finally, Objective-C. In 2001, he founded Random Ideas, LLC, a software company dedicated to building great applications for the Mac. When the iPhone SDK was announced, he was honored to be selected as one of the earliest developers to have access to the SDK, and using it, he developed several applications that were available in the iTunes App Store when it launched on July 11, 2008. Since that time, his iPhone applications have received several awards — including being chosen as among the Top 100 apps and games in the App Store — and been featured as Apple Staff Picks and in Apple advertisements. Today, Jiva continues to work full time for his company, developing great iPhone and Mac applications. He lives with his wife, three children, and two basset hounds in the desert of Arizona.



# Credits

## **Acquisitions Editor**

Aaron Black

## **Executive Editor**

Jody Lefever

## **Project Editor**

Martin V. Minner

## **Technical Editor**

Dallas Brown

## **Copy Editor**

Marylouise Wiack

## **Editorial Director**

Robyn Siesky

## **Editorial Manager**

Cricket Kregel

## **Business Manager**

Amy Knies

## **Senior Marketing Manager**

Sandy Smith

## **Vice President and Executive Group Publisher**

Richard Swadley

## **Vice President and Executive Publisher**

Barry Pruett

## **Project Coordinator**

Katie Crocker

## **Graphics and Production Specialists**

Andrea Hornberger

Jennifer Mayberry

Mark Pinto

## **Quality Control Technician**

Rebecca Denoncour

## **Proofreading**

Christine Sabooni

## **Indexing**

BIM Indexing & Proofreading

## **Media Development Project Manager**

Laura Moss

## **Media Development Assistant Project Manager**

Jenny Swisher



# Contents

Preface .....	xix
Acknowledgments.....	xxi
<b>Part I: Getting Started with the iPhone.....</b>	<b>1</b>
<b>Chapter 1: Getting to Know Your Tools.....</b>	<b>3</b>
Introducing Xcode .....	3
Introducing Interface Builder.....	5
Summary.....	6
<b>Chapter 2: Building a Basic iPhone Application .....</b>	<b>7</b>
Creating a Project from a Template .....	7
Building a Simple GUI.....	11
Writing a Simple UIViewController.....	14
Launching on the iPhone.....	16
Summary.....	18
<b>Chapter 3: Exploring the Simulator in Depth .....</b>	<b>19</b>
Exploring the Simulator UI .....	19
Simulating multi-touch .....	20
Simulating Core Location .....	21
Turning the simulator on its side .....	23
Digging deeper into the simulator filesystem.....	23
Understanding the Differences between the Simulator and the Real Thing .....	25
Avoiding APIs that won't work on a real device.....	25
Understanding performance differences .....	26
Summary.....	27
<b>Part II: Building GUIs.....</b>	<b>29</b>
<b>Chapter 4: Understanding the Basics of an iPhone User Interface .....</b>	<b>31</b>
Looking at the iPhone Screen .....	32
Creating a Cinematic User Experience.....	33
Looking at the Built-in Applications .....	34
Understanding the Basics of iPhone Navigation .....	36
Summary.....	37

<b>Chapter 5: Introducing UIViewController . . . . .</b>	<b>39</b>
Understanding the Basics of Model, View, Controller . . . . .	40
Exploring UIViewController . . . . .	41
Overriding Methods on UIViewController . . . . .	41
Handling view rotations . . . . .	43
Handling memory warnings . . . . .	44
Working with UIViewController . . . . .	44
Creating the code . . . . .	45
Creating the UIViewController header . . . . .	46
Creating the UIViewController implementation . . . . .	47
Adding the controller to Interface Builder . . . . .	51
Understanding the UIApplicationDelegate . . . . .	53
Creating the model in the App Delegate . . . . .	53
Adding the view to the window . . . . .	55
Summary . . . . .	57
<b>Chapter 6: Using UITableView . . . . .</b>	<b>59</b>
Understanding UITableView Data Sources and Delegates . . . . .	59
Understanding NSIndexPath and how it works with UITableViews . . . . .	60
Exploring UITableViewDataSource . . . . .	60
Exploring UITableViewDelegate . . . . .	64
Adding a UITableView to an Application . . . . .	66
Making a UITableViewDataSource . . . . .	67
Seeing the basic application run . . . . .	73
Taking action when a row is touched . . . . .	75
Configuring a UITableView to Be Grouped . . . . .	78
Adding the state to your model . . . . .	78
Updating your UITableViewDataSource . . . . .	79
Doing Advanced UITableView Configuration . . . . .	83
Adding an alphabetical list to the side . . . . .	83
Adding search . . . . .	84
Summary . . . . .	84
<b>Chapter 7: Working with UITableViewCell . . . . .</b>	<b>85</b>
Understanding the Parts of a UITableViewCell . . . . .	85
Adding Images to a UITableViewCell . . . . .	87
Performing Deeper Customization of UITableViewCell . . . . .	90
Thinking about Performance in Custom Cells . . . . .	94
Reusing UITableViewCell . . . . .	96
Understanding the reuse identifier . . . . .	97
Summary . . . . .	99
<b>Chapter 8: Working within the UINavigationController Model . . . . .</b>	<b>101</b>
Understanding the UINavigationController Navigational Model . . . . .	101
Creating a UINavigationController . . . . .	102
Configuring a UINavigationController . . . . .	105

Pushing and Popping UIViewControllers .....	106
Adding a UIToolbar .....	107
Allowing Editing of the Rows .....	107
Moving MyView into a UINavigationController .....	107
Making Your EditViewController .....	109
Editing rows .....	111
Adding rows .....	115
Deleting rows .....	116
Summary .....	116
<b>Chapter 9: Exploring the Purpose of the UIApplicationDelegate .....</b>	<b>117</b>
Handling Startup and Shutdown .....	118
Understanding launch methods .....	118
Understanding the applicationWillTerminate method .....	119
Receiving Notifications while Your Application Is Running .....	120
Handling resource alerts .....	121
Handling phone calls and sleep .....	122
Changing status .....	122
Handling remote notifications .....	123
Summary .....	123
<b>Chapter 10: Applying Custom UIViews .....</b>	<b>125</b>
Understanding Cocoa Touch View Geometry .....	125
Adding Custom Views to a Project .....	126
Implementing the Custom View Code .....	127
Working with Core Graphics .....	128
Implementing drawRect .....	133
Summary .....	136
<b>Chapter 11: Handling Touch Events .....</b>	<b>137</b>
Handling Touch Events in a Custom UIView .....	137
Implementing touchesBegan:withEvent: .....	138
Working with the touches NSSet .....	139
Implementing touchesMoved:withEvent: .....	140
Implementing touchesEnded:withEvent: .....	140
Handling multi-touch events .....	140
Updating Your Custom View with Touch Events .....	140
Moving the circle with a touch .....	141
Adding scaling .....	144
Summary .....	146
<b>Chapter 12: Working with Other Controls .....</b>	<b>147</b>
Finding What Other Controls Are Available in Cocoa Touch .....	147
Working with a UISlider .....	148
Configuring a UISlider through Interface Builder .....	148
Updating the status of your UISlider .....	151

Using UITabBar .....	151
Configuring a UITabBar through Interface Builder .....	153
Adding UITabBarItem to a UITabBar .....	153
Doing advanced configuration of UITabBars .....	155
Using UIPickerView .....	155
Configuring UIPickerView through Interface Builder .....	155
Creating the UIPickerViewDataSource .....	156
Creating a UIPickerView delegate .....	157
Summary .....	157
<b>Chapter 13: Handling Text Input .....</b>	<b>159</b>
Configuring the Keyboard through Interface Builder .....	159
Configuring capitalization .....	161
Enabling and disabling auto correction .....	162
Setting the keyboard type .....	162
Setting the behavior of the Return key .....	163
Looking at other settings .....	163
Working with a Keyboard through Code .....	163
Making the keyboard appear .....	163
Understanding UITextInputTraits .....	164
Handling Events for the Keyboard .....	165
Creating a UITextFieldDelegate .....	166
Moving your view in response to the keyboard appearance .....	167
Summary .....	169
<b>Chapter 14: Building Cinematic UIs with Core Animation .....</b>	<b>171</b>
Using the UIViews Animation Methods .....	171
Using Advanced Core Animation with CALayer .....	178
Animating UIImageViews .....	181
Summary .....	182
<b>Chapter 15: Using OpenGL ES .....</b>	<b>183</b>
Understanding the Capabilities of iPhone OpenGL ES .....	183
Creating a Very Basic OpenGL View .....	184
Summary .....	188
<b>Chapter 16: Integrating Safari .....</b>	<b>189</b>
Opening URLs Using the iPhone Web Browser .....	189
Using UIWebView .....	190
Adding a UIWebView to your application .....	191
Loading a URL .....	191
Implementing a UIWebViewDelegate .....	192
Loading HTML content from the application bundle .....	193
Summary .....	193

<b>Part III: Working with Data .....</b>	<b>195</b>
<b>Chapter 17: Storing User Defaults .....</b>	<b>197</b>
Acquiring the UserDefaults Object.....	197
Reading and Writing Values to UserDefaults.....	198
Setting Your Default Defaults .....	201
Using the Settings App .....	203
Adding a settings bundle.....	203
Adding settings to your settings bundle.....	204
Summary.....	207
<b>Chapter 18: Implementing a Database with Core Data .....</b>	<b>209</b>
Understanding Core Data's Building Blocks .....	210
Adding Core Data Support to Favorite Cities .....	211
Modifying the app delegate .....	212
Working with the Managed Object Model .....	216
CRUD — Creating, Reading, Updating, Deleting.....	219
Creating.....	220
Reading.....	220
Updating .....	221
Deleting.....	222
Bringing it together and updating your app .....	222
Understanding What Core Data Makes Easier .....	230
Understanding What Core Data Is Not Good For .....	230
Summary.....	231
<b>Chapter 19: Connecting to the World with Networking.....</b>	<b>233</b>
Accessing the Web.....	234
Using URLs with foundational classes.....	234
Using NSURLRequest and NSURLConnection.....	235
Sending E-mail from within Your App with the Message UI Framework .....	239
Using Core Foundation Sockets.....	243
Exploring CFSocket .....	244
Getting host info with CFHost.....	247
Using CFStreams.....	249
Exploring Bonjour.....	252
Browsing for Bonjour services .....	253
Using NSNetServices.....	255
Summary.....	256
<b>Chapter 20: Using the Push Notification Service .....</b>	<b>257</b>
Understanding the Push Notification Service Architecture .....	258
Understanding the push notification communications.....	259
Understanding push notification security.....	264

Acquiring Push Notification Certificates.....	266
Developing Your Server-Side Push Notification Service .....	269
Implementing a Ruby push notification supplier.....	269
Pushing notifications.....	272
Checking delivery using feedback.....	272
Integrating Push Notifications with Your iPhone Client.....	273
Registering for notifications .....	273
Receiving notifications in your application .....	274
Summary.....	275
<b>Chapter 21: Using the Game Kit API.....</b>	<b>277</b>
Providing Peer-to-Peer Connectivity .....	277
Finding peers.....	278
Working with sessions.....	281
Providing In-Game Voice.....	284
Summary.....	288
<b>Chapter 22: Implementing Cut, Copy, and Paste .....</b>	<b>289</b>
Copying and Pasting with Standard Controls.....	289
Understanding Pasteboard Types.....	290
Interacting with UIPasteboard .....	291
Implementing Cut, Copy, and Paste on a Custom View .....	292
Implementing touchesEnded:withEvent: to display the menu.....	292
Implementing the copy: method.....	293
Implementing the paste: method .....	294
Understanding the interactions.....	294
Summary.....	295
<b>Chapter 23: Using the Maps API .....</b>	<b>297</b>
Showing an Embedded Map with MKMapView.....	297
Creating an MKMapView .....	297
Specifying the map region.....	299
Annotating Maps.....	301
Thinking about performance with annotations .....	303
Converting Coordinates .....	303
Summary.....	305
<b>Part IV: Working with Media .....</b>	<b>307</b>
<b>Chapter 24: Exploring the Supported Media Types.....</b>	<b>309</b>
Supported Audio Formats.....	309
Compressed audio .....	309
Uncompressed audio.....	310
Supported Video Formats.....	310
Summary.....	310



<b>Chapter 25: Playing Audio .....</b>	<b>311</b>
Using the AV Foundation Framework .....	311
Setting your audio configuration with AVAudioSession .....	311
Using an AVAudioSessionDelegate .....	314
Playing audio with AVAudioPlayer .....	315
Using an AVAudioPlayerDelegate .....	317
Playing Audio with OpenAL .....	317
Summary .....	318
<b>Chapter 26: Accessing the iPod Library. ....</b>	<b>319</b>
Working with the Media Player Framework .....	319
Accessing the Media Library .....	320
Using the MPMediaPickerController .....	320
Searching for media .....	323
Working with Player Controllers .....	327
Accessing Media Artwork .....	328
Summary .....	328
<b>Chapter 27: Recording Audio .....</b>	<b>329</b>
Setting up Your AVAudioSession .....	329
Allocating an AVAudioRecorder .....	330
Creating a Voice Recorder .....	332
Summary .....	337
<b>Chapter 28: Playing Video in Your Application .....</b>	<b>339</b>
Playing Video Files Contained in the App Bundle .....	339
Playing Video from the Internet .....	341
Summary .....	342
<b>Part V: Working with the iPhone Hardware .....</b>	<b>343</b>
<b>Chapter 29: Discovering Information about the Device. ....</b>	<b>345</b>
Accessing the Battery State .....	345
Accessing the Proximity Sensor .....	346
Working with Device Metadata .....	346
Summary .....	347
<b>Chapter 30: Getting Your Location Using Core Location .....</b>	<b>349</b>
Finding Where You Are .....	349
Allocating a CLLocationManager .....	351
Setting yourself as the delegate .....	352
Implementing the CLLocationManagerDelegate protocol .....	352
Starting and stopping location updates .....	353
Viewing your location on Google Maps .....	354

Narrowing the Accuracy of the Coordinates.....	356
Filtering Location Updates.....	356
Looking at the Final Code .....	357
Working with the iPhone 3GS Compass.....	359
Summary.....	360
<b>Chapter 31: Working with the Accelerometer.....</b>	<b>361</b>
Determining Which Way Is Up.....	361
Building a bubble level.....	364
Understanding the UIAcceleration object .....	367
Capturing Shake Events.....	367
Building a “shake to break” detector .....	368
Implementing drawRect .....	368
Implementing motionBegan and motionEnded.....	370
Cancellation of motion events .....	370
Summary.....	371
<b>Chapter 32: Interfacing with Peripherals .....</b>	<b>373</b>
Creating Accessories That Interface with iPhone .....	373
Finding Accessories That Are Connected Using EAAccessoryManager.....	374
Understanding the EAAccessory Class.....	375
Working with EASession.....	375
Talking to Your Device Using NSSStreams .....	376
Summary.....	377
<b>Part VI: Handling Distribution.....</b>	<b>379</b>
<b>Chapter 33: Code Signing Your Apps.....</b>	<b>381</b>
Acquiring a Development Certificate .....	381
Understanding the Provisioning Process .....	382
Understanding development, distribution, and ad hoc .....	382
Installing provisioning profiles .....	383
Exploring what happens when an app is signed.....	384
Setting up your build to be signed.....	385
Doing Ad Hoc Builds .....	386
Configuring the build in Xcode .....	386
Distributing to users .....	388
Summary.....	388
<b>Chapter 34: Expanding Your Application Using the In-App Purchase Support .....</b>	<b>389</b>
Knowing the Types of Things You Can Sell.....	389
Working with Unlockable Content.....	390
Setting up Purchasable Content in iTunes Connect.....	391

Working with Store Kit .....	392
Verifying app purchase availability.....	393
Presenting your store .....	393
Making the purchase .....	395
Processing the payment .....	395
Verifying the transaction .....	397
Unlocking the content .....	397
Restoring purchased content.....	398
Understanding In-App Purchasing testing .....	399
Summary.....	399
<b>Index.....</b>	<b>.401</b>



# Preface

**M**y goal in writing this book was to provide a comprehensive toolkit for both new and experienced iPhone developers. Its focus is intended to be primarily on the new technologies of iPhone OS 3, but it includes enough general iPhone development material that even a new developer to the platform will receive a great deal of benefit from reading it.

Writing a book about Cocoa Touch programming is an incredible challenge. It's very difficult to judge the technical capability of the typical reader of a book like this. Does the reader already know Objective-C? Has he or she already been developing for Mac OS X? These are the first questions that I had to ask myself when I began to work on this project.

In the end, I decided that with the development of iPhone OS 3, it was a unique opportunity where the new and experienced reader intersected and therefore, I thought it was an ideal time to write a book that would be useful to them both.

So I decided that this book would contain some introductory material, but that it would primarily focus on the new features of iPhone OS 3. In this way, it provides an excellent bridge for learning the technologies of the new operating system, as well as an introduction to general iPhone development.

As a reader, you are expected to already be somewhat familiar with Objective-C. Additionally, you should be somewhat familiar with either Mac OS X, or iPhone development, though you need not be an expert.

If you're completely new to the platform, I suggest picking up a book specifically on Objective-C to complement this book. There are several such books on the market, but I recommend one that approaches Objective-C from the point of view of learning the core language itself, rather than one that mixes an introduction to Objective-C with other topics. After you've worked through enough of that book to feel comfortable with the syntax of the language, you can begin this book at Chapter 1.

If you've already been developing for Mac OS X, but haven't done any iPhone development, you probably already know Objective-C and are familiar with many of the tools used in iPhone development. However, while iPhone development is similar to Mac OS X development, it's not exactly the same. So I suggest that you skip Chapter 1, and start reading Chapter 2, where you build a basic iPhone app from scratch.

Finally, if you're already an experienced iPhone developer, you already know all the basics involved in iPhone development; you're just here for the iPhone OS 3 material. I suggest you start at Chapter 5. All the chapters were written with the new and updated iPhone OS 3 API in mind, and so even if you are already familiar with UIViewController and friends, you will find new material there.

In these ways, this book provides an excellent extension to your library if you already have books on any of these three subjects.

With regard to conventions used within this book, I've tried to be reasonably consistent, and also tried to generally err on the side of Apple conventions when prudent. The only notable exception has been in my use of the term "method" to indicate functions on instances and

classes. Apple generally prefers the term *message*. This is in part due to the influence of Smalltalk on Objective-C.

Also with regard to method calling conventions, Objective-C is known for being particularly verbose in its method names. As a result, some abbreviation conventions have been adopted when writing about particular methods. I have chosen to follow two standards. The first, I use when the usual use of the method in question is in implementing your own, or in overriding it in a subclass. This is most commonly the case when dealing with delegate methods. In these cases, because it's helpful to know the entire signature of the method, in order to write your own implementation, I have chosen to include the entire method signature. So, for example, these will be written as `-(BOOL)foo:(NSString *)bar withBaz:(NSString *)baz`. As you can see, in this example, we have a method called `foo`, which takes two parameters of type `NSString *`, one called `bar` and a second called `baz`, and which returns a `BOOL` value.

Alternatively, in the case where your typical use will be only to use the method in your own code, since Xcode generally automatically completes the types of the parameters for you, I have chosen to use the abbreviated form of writing the method signatures. In the case of the previous method, that means it's written `foo:withBaz:`. Notice that wherever a parameter is inserted, a colon holds the place of the parameter.

When referring to keyboard shortcuts, I opted to use the term *Command key* or the  $\text{⌘}$  symbol to indicate keyboard shortcuts using the key directly to the left of the space bar on most Apple keyboards. You may also know this as the "Apple" key, as until only a few years ago, it included a small Apple logo on it. Additionally, the key next to the Command key has been called the *Option key* and the key next to that, the *Control key*. These should all be consistent with Apple documentation conventions.

When discussing the use of menus in Xcode, I've used the technique of separating the nested menu items using arrow notation. So, to describe the *New File* sub item of the *File* menu, it will be written as *File* ⇨ *New File*.

Finally, with regard to sample code, in chapters where I have instructed you to build specific full projects, I have generally tried to include full listings for the code. In cases where I have not, you can always download the projects, complete with artwork and other supporting files, from the book's Web site, located at <http://www.wiley.com/go/cocoatouchdevref>. There are also chapters where it didn't really make sense to create a full project to demonstrate a technology. In these cases, the code listings are snippets that you can use as a basis for your own code. Because these snippets don't comprise fully functional projects, there will not be example projects for them on the Web site.

I hope that you find this book as enjoyable an experience to read as I had writing it. To me, the mark of a good technical book is that it doesn't sit on my shelf. It holds a place of honor by my desk because I keep returning to it, time and again. I hope that this book holds such prestige in your library, and that it becomes a dog-eared, cover-torn, page-scribbled-on reference that remains useful to you for years to come.

Jiva DeVoe

book@random-ideas.net

# Acknowledgments

Writing this book has been one of the most challenging and exciting projects I have done in my career, but I could not have done it without the aid and support of some specific individuals whom I would like to thank.

First, I'd like to thank my friend and technical editor, Dallas Brown, of HashBang Industries, who took up the challenge of correcting my mistakes and keeping me honest. Your time spent, and your thoughtful comments were excellent.

Along that same line, I'd also like to thank my friend, Brad Miller, of Cynical Peak Software, who also provided welcome criticism and an extra set of eyes on several chapters, and who always seems to be awake and online at the same times I am.

For pressing forward with the book, even with a tight schedule, I'd like to thank all the folks at John Wiley & Sons. I look forward to working with you on new projects in the future.

For teaching me to marvel at the wonders of technology and encouraging me to pursue my dreams in computers, I'm thankful to my father, Robert A. DeVoe.

Many thanks to my children, who have endured these several months of my working late without complaint. You have earned your trip to Disneyland! It is for you that I do everything I do.

Finally, and most importantly, I'd like to thank my wife, for her unerring support, not just in this project, but in all my work. Without her, this book could not have been finished. You lift me when my spirits are low and tired, and inspire me to keep reaching for new accomplishments and goals. I can't thank you enough.








# Getting Started with the iPhone



 <b>In This Part</b>
Chapter 1 Getting to Know Your Tools
Chapter 2 Building a Basic iPhone Application
Chapter 3 Exploring the Simulator in Depth



# 1

# Getting to Know Your Tools

They say that when a craftsman finds a tool that he loves, over time it becomes an extension of him. He learns its idiosyncrasies inside and out, backwards and forwards, and this leads to a relationship that transcends simple use of the tool and instead becomes more involved. The tool becomes an extension of his hand, an extension that he can guide with an almost extrasensory vision.

You are fortunate then, that Apple provides you with an excellent set of free tools for developing software for the iPhone. They are tools that have evolved over the last 15 years of Objective-C development, first on NeXT computers, then on Mac OS X, and now for iPhone.

In this chapter, you will take a brief look at these tools and learn where you can find more information about them. They are incredibly powerful tools that seem to be unique in software development, both for their ability to provide enough power to enable incredibly complex software systems to be developed, and also because they seem to know just when to stay out of your way and simply provide a great text-editing environment for you to write code in.

Unfortunately, a comprehensive description of every last feature of these applications is beyond the scope of this book, and so I won't be delving into them in great detail. My main goal here is simply to introduce you to the tools so that you're familiar with them.

## Introducing Xcode

The cornerstone of iPhone software development is the Xcode integrated development environment, or IDE. Xcode originated on NeXT Step computers as Project Builder. Over the years, it has gone through many revisions to finally arrive at the version that is available to you today. It uses GCC as its underlying compiler technology and provides many sophisticated features found in modern IDEs today, such as code completion, re-factoring, and sophisticated code navigation. Interestingly, it also has one of the best cross-platform compiling capabilities of any modern IDE. With it, you can compile for Intel, PowerPC, iPhone OS, or even (with third-party tools) Microsoft Windows. You can do all of this simply by configuring targets in the IDE.

1

**In This Chapter**

Becoming familiar with  
Xcode and Interface  
Builder

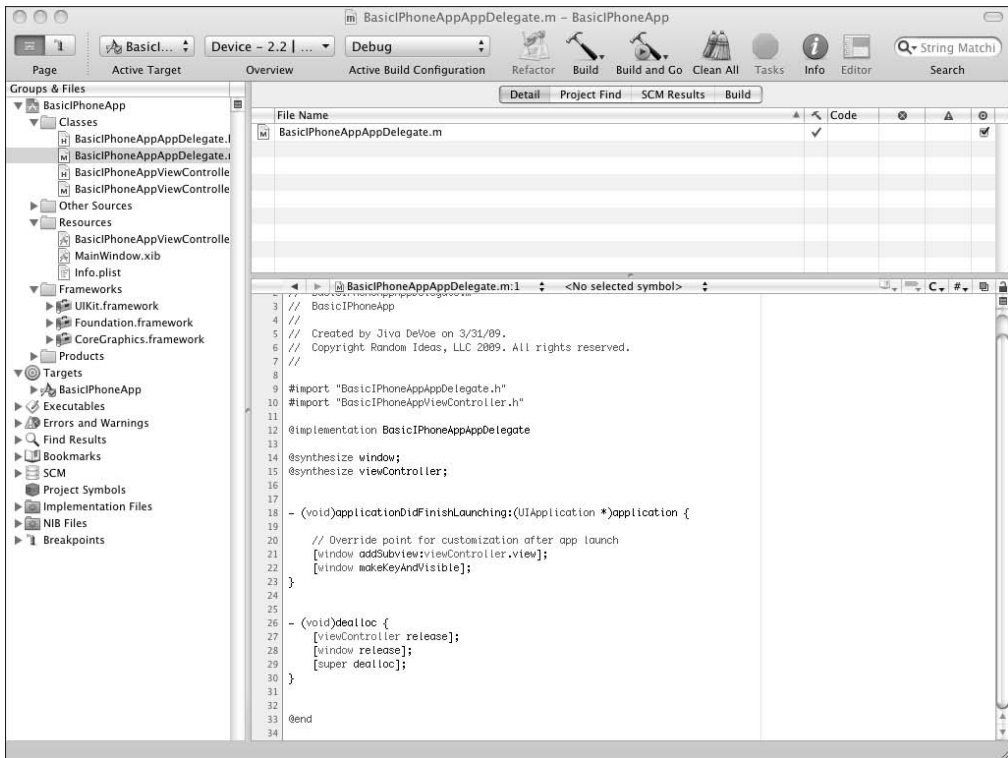
**NOTE**

To download Xcode, all you need to do is sign up for a free developer account on the Apple Web site. You can do this at <http://developer.apple.com/iphone>.

Figure 1.1 shows the main Xcode window. In it, you can see the left panel, which shows the file organization view. From here, you can drag and drop files into your projects or organize them by groups. It also provides the ability to organize your files by Smart groups, which are built using search queries and can be useful for looking for particular files in your projects.

**Figure 1.1**

The Xcode interface



On the right side, you can see the main editing window. It is in this window that you will do the majority of your programming work. At the top of the text-editing window, you can see some

drop-down menus that enable you to quickly jump to any recently opened files. The second drop-down menu from the left enables you to quickly jump to any method in the current file. Holding down the  $\mathbb{C}$  key and using the right and left arrow keys enables you to quickly navigate backwards and forwards through the file history. Additionally, holding down the Option and  $\mathbb{C}$  keys together and pressing the up arrow key enables you to quickly swap between the implementation and header files for the currently active compilation unit.

The Xcode editor is quite sophisticated and can be configured with a variety of shortcuts and hot-keys that make your editing much easier and faster. For example, using the Option key and the right and left arrow keys enables you to quickly jump from word to word in your code. Holding down the  $\mathbb{C}$  key and using the left and right arrow keys enables you to quickly jump to the front or beginning of the current line.

It's a good idea to learn the intricacies of the Xcode editor inside and out, because it is such a fundamental tool to everything that you will be doing as an iPhone OS developer.

## Introducing Interface Builder

The second major component of the Xcode programming environment is the graphical user interface (GUI) builder called Interface Builder, shown in Figure 1.2. It is this application that you will use to draw your GUI for your application and connect your buttons to actions in your code.

Many developers coming to iPhone development from other environments are sometimes confused by Interface Builder because it doesn't generate any code. However, this is an asset and not a liability. IDEs that simply generate code tend to be more difficult to work with over time, as the code that they generate becomes out of sync with the user interface. Interface Builder uses more of a metadata style approach. This means that you tell it that you want to instantiate an object of a given type, and when your nib is loaded, it goes and finds the class for that type, instantiates it, and attaches the outlets and actions that you have configured to the appropriate places. It does not serialize actual instances of your objects, nor does it generate code that is compiled.

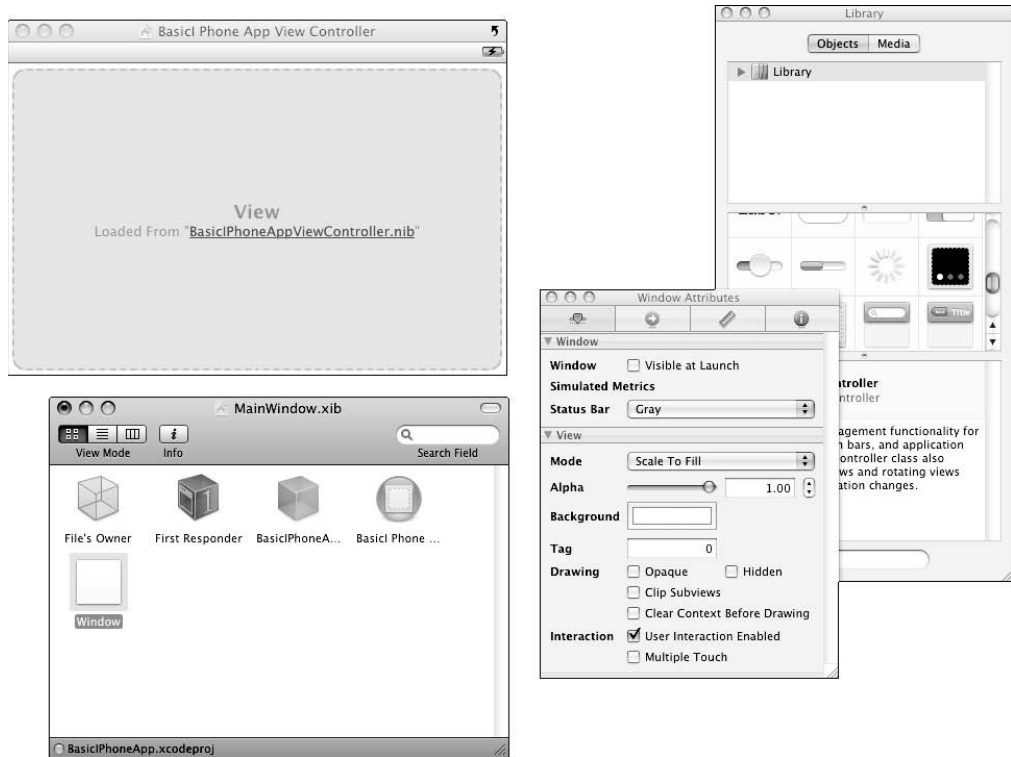


### NOTE

The files that Interface Builder saves are referred to as nib files. This stands for NeXT Interface Builder. The file extension of the nib files used with the iPhone is `.xib`. This is to differentiate them as containing XML versus the older, original NeXT format.

I will talk about Interface Builder in a bit more detail in Part II. For now, the important thing to know is that you can start Interface Builder either separately by simply launching it or by double-clicking any of the `.xib` files in your project.

**Figure 1.2**  
Interface Builder



## Summary

In this chapter, I introduced you to Xcode and Interface Builder. In the upcoming chapters, you will use these tools in much more depth, but I wanted to simply give you a brief overview of what they look like so that when you use them in the future you will be familiar with them. I encourage you to read through the documentation on them very carefully and try to become as intimately familiar with them as you can.