Extracted from:

iOS SDK Development

This PDF file contains pages extracted from *iOS SDK Development*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit http://www.pragprog.com.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printer versions; the content is otherwise identical.

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina

The Pragmatic Programmers

iOS SDK Development

Chris Adamson
and Bill Dudney

Edited by Brian P. Hogan

iOS SDK Development

Chris Adamson
Bill Dudney

The Pragmatic Bookshelf
Dallas, Texas • Raleigh, North Carolina

Pragmatic
Bookshelf

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

The team that produced this book includes:

Brian P. Hogan (editor)
Potomac Indexing, LLC (indexer)
Molly McBeath (copyeditor)
David J Kelly (typesetter)
Janet Furlow (producer)
Juliet Benda (rights)
Ellie Callahan (support)

Printed in the United States of America.
ISBN-13: 978-1-934356-94-4
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—November 2012

Introduction

When we wrote the first edition of *iPhone SDK Development*, we knew we weren't writing something like *The C Programming Language* that would live on unchanged for decades. But was two years on the shelf too much to ask for?

Apparently so! And we're not complaining. Since the release of our first book in early 2009, the platform has surged in popularity and prominence. In the intervening years, the platform has added a whole new device family in the iPad, sold millions of units, and changed its name from *iPhone OS* to *iOS* to better reflect its multiple uses and perhaps to leave the door open to future devices.

The SDK has also grown in breadth and depth, adding new features, new frameworks, and new tools. Since the first book, Apple has changed compilers and has radically overhauled Xcode, the primary iOS development environment. As we did our day-to-day work with all this new stuff, we'd sometimes look over the book and notice each time that more and more of it was out-of-date. The Xcode screenshots, the callbacks to selectors instead of blocks, the exposure of private variables and method names in the public header—all this stuff we weren't using ourselves anymore—it all seemed so…so *2009*.

A New Start and a Do-Over

If you happen to have read the previous book and then you flip through this one, you'll notice something: we have copied over *absolutely nothing* from the old book. This one is 100 percent new. As we looked at all the changes to the platform—between Xcode 4, iOS 6, and the iPad—we decided that so much had changed that we would be better off starting off fresh. This freed us to embrace everything that's new, making a complete cut with the past and writing a truly up-to-date book.

And there's so much that's new! The radically overhauled Xcode 4 is the first version of the development tool that's truly built for iOS development, rather than having iPhone concepts bolted onto a Mac IDE. It completely rethinks how developers work with projects, and its sensible conceptual divisions make finding its functionality more predictable. In code, the revolutionary new Automatic Reference Counting frees developers from the drudgery of manual retain and release calls, a routine that if mishandled would lead to memory leaks or crashes. With multicore processors like the A5 comes a need for practical concurrent programming, something the iOS SDK answers with Grand Central Dispatch, a technology that allows programmers to divvy up small bits of code and data as “blocks” and let the system decide how best to run them.

In fact, it’s possible to have too much of a good thing, and the iOS 6 SDK is a good example. In our first book, we worked to present most of the interesting things you could do with the platform and watched as our 250-page book grew to nearly 600 pages, blowing through deadline after deadline. And that was just for iPhone OS 3. To cover all the subsequent changes in iOS versions 4 through 6 at the same depth, we’d be in the thousands of pages. And that’s not very “pragmatic.”

So we’ve adjusted our focus for this edition. This book is about setting you off on the right foot: understanding the fundamentals, getting comfortable with the tools and the concepts, and developing good habits. We’ve put a particular emphasis on the last of these, looking for the kinds of things that aren’t just handy classes or compiler tricks but instead are the values and routines that will help produce better apps. To that point, you’ll find we spend time talking about topics like internationalization, testing, debugging, and source code management. We’re also adopting modern iOS development practices, such as using Objective-C properties exclusively instead of using traditional instance variables and getting private methods out of public header files.

Our goal is for this book to serve as a prerequisite for all the other iOS titles from the Pragmatic Bookshelf, such as *iOS Recipes: Tips and Tricks for Awesome iPhone and iPad Apps [WD11]* by Paul Warren and Matt Drance and *Core Data [Zar12]* by Marcus S. Zarra. And, of course, it should provide a good grounding for any future titles that dig further into the many frameworks of iOS.

But more importantly, you should come away from this book with a firm grasp of the most essential iOS APIs—the UIKit GUI framework and the essential utilities of the Foundation framework—and enough of a sense of where things are and how things work to be able to grab the documentation for interesting-looking features and be able to figure it out.

So Here's the Plan

With that goal in mind, let's look at how we're going to get there. We'll start by getting Xcode from the Mac App Store, and by the time we're done we'll be ready to upload our own apps to the iOS App Store. Here's a road map to the journey:

- Chapter 1, *Tweetings and Welcome to iOS 6*, on page ?, starts by downloading and installing the SDK and beginning work on a first app, which uses iOS 6's new Social framework to send a tweet telling the world that our journey is underway. We'll use Xcode's visual tools to build a user interface and connect it to our first code.

- Chapter 2, *Programming for iOS*, on page ?, gets into the specifics of coding by introducing the Objective-C programming language and the two frameworks we use most frequently in iOS apps: Foundation and UIKit.

- Chapter 3, *Asynchronicity and Concurrency*, on page ?, addresses the issues of how and when our code is run, showing how many of the iOS APIs use asynchronous callbacks and employing the Grand Central Dispatch system to handle concurrent execution of our code.

- Chapter 4, *View Controllers*, on page ?, turns our attention back to the UI and looks at how iOS apps are built on a strong Model-View-Controller (MVC) foundation, which will let us make our code more resilient and easier to maintain.

- Chapter 5, *Table Views*, on page ?, continues to build our UI arsenal by bringing in the flexible and widely used table view, the linchpin of most iPhone apps that need to present lists of data.

- Chapter 6, *Storyboards and Container Controllers*, on page ?, is where we'll learn how to build a visual road map of the many screens of an app and how to build much of the logic of that navigation and presentation automatically.

- Chapter 7, *Documents and iCloud*, on page ?, gives us the tools we need to save our user's work to the filesystem as well as to Apple's new iCloud service.

- Chapter 8, *Drawing and Animating*, on page ?, lets us bring our own pixels to the game using the Core Graphics framework to draw images and shapes and use Core Animation to give them life.

- Chapter 9, *Testing and Fixing Apps*, on page ?, addresses the things that can go wrong when we build and run our apps and how we use the SDK's tools both to make them right and to make sure they don't go wrong again.

- Chapter 10, *The App Store and Beyond*, on page ?, completes our journey by changing our outlook from learning to doing. We'll start maintaining our code for the long haul, running it on the device, submitting it on the store (without getting rejected), and managing it after it's in users' hands.

Expectations and Technical Requirements

The technical requirements for iOS development, in general terms, are pretty simple: a reasonably new Mac, running the most-recent production version of Mac OS X. The specific version numbers increment ever upward; check out Xcode on the Mac App Store for the latest requirements. For this edition, our baseline is Xcode 4.5 and the iOS 6 SDK (included with Xcode 4.5), running on Mountain Lion (10.8.2).

We also expect readers of this book to be proficient programmers in at least one object-oriented language. That can be one of the many curly-brace descendants of C (C++, C#, or Java), or an OO scripting language like Ruby or Python. In the previous edition, we assumed some previous familiarity with C and its memory-management concepts (pointers, `malloc()`, and so on), but we found many readers didn't have it. For this edition, we are providing a catch-up appendix for readers who've never had to master these challenges. If the * and & memory operators are unfamiliar, or perhaps terrifying, Appendix 1, *Wait! I Forgot (or Never Learned) C!*, on page ?, will lay out the C essentials needed to work with Objective-C, the primary language of iOS development.

Online Resources

This book isn't just about static words on a page or screen. It comes with a web page, http://www.pragprog.com/titles/adios, where you can learn more and access useful resources:

- Download the complete source code for all the code examples in the book as ready-to-build Xcode projects.

- Participate in a discussion forum with other readers, fellow developers, and the authors.

- Help improve the book by reporting errata, such as content suggestions and typos.

If you're reading the ebook, you can also access the source file for any code listing by clicking on the gray-green rectangle before the listing.

And So It Begins

We're now ready to begin digging into the iOS 6 SDK. In the next chapter, we'll tool up, familiarize ourselves with the development environment, and write our first app. We'll revise this app over the course of the first few chapters as our skills grow and we learn new tricks.

Anytime you get stuck, check against the source code from the book's page or join us in the forum to let us know what's going on.

Let's go!