

Extracted from:

iPhone SDK Development

Building iPhone Applications

This PDF file contains pages extracted from iPhone 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.

Copyright © 2009 The Pragmatic Programmers, LLC.

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, without the prior consent of the publisher.

The Pragmatic
Programmers

Add Book

Save

The Pragmatic
Programmers

iPhone SDK Development |

Bill Dudney and
Chris Adamson

Edited by Daniel H Steinberg





Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

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.

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at

<http://www.pragprog.com>

Copyright © 2009 Bill Dudley and Chris Adamson.

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, without the prior consent of the publisher.

Printed in the United States of America.

ISBN-10: 1-934356-25-5

ISBN-13: 978-1-934356-25-8

Printed on acid-free paper.

P2.0 printing, December 2009

Version: 2010-2-3

Chapter 1

Introduction

The iPhone has changed everything.

Maybe by the time you read this you'll no longer notice that the latest cool new phone from every manufacturer looks and behaves more like the iPhone than like the models they used to sell before the iPhone was released. Meanwhile, at the time we write this, Apple's iPhone ads showcase the apps that you and other third-party developers are creating. The Apple marketing campaign is highlighting the iPhone as a platform for great software—software that you will create.

It's hard to overstate just how much the iPhone SDK and the App Store have changed the way we develop, release, and consume mobile software, even in the short time that they've been available.

Before the iPhone, writing software for the small device was rife with hazards. Many developers complained of *fragmentation*, the way that different devices would have different characteristics (screen sizes, color depths, input technologies, and so on) and different implementations of supposedly standard technologies, whose variant behaviors forced developers into an expensive and difficult game of “write once, test everywhere.” And that's only in the cases where they had access to a market; in some cases, handset makers and mobile telephony carriers used the technology vendors' security practices as a means of enforcing business models, allowing only first-party applications on the device or crippling third-party apps by denying essential services, such as access to the network. It was enough to send developers fleeing back to the desktop, if not the server.

As an iPhone SDK developer, you don't need to worry about fragmentation. The iPhone and iPod touch are highly predictable, with only a few models thus far and highly consistent features and behavior between them. The SDK brings the tools and technologies of Mac OS X—refined by years of use creating world-class desktop applications—and makes its essential elements available for creating iPhone and iPod touch applications. While other mobile platforms throw an exception if you touch the networking stack, the iPhone gives you exceptional technologies such as the self-networking Bonjour. On other devices, playing a sound file is a hit-or-miss affair, while the iPhone offers extraordinarily deep support for media. And when you've finished writing your application, instead of a carrier demanding that you “partner” with them (and give them 99 cents of every dollar you make) and instead of being shut out entirely if you're not a big company touting a well-known intellectual property, the App Store lets you put your application before every iPhone and iPod touch user. And you set your own price with Apple taking a much smaller and quite reasonable cut.

The result has been revolutionary. At the rate the App Store is growing, there's no point quoting numbers because they'll be completely out-of-date by the time you read this book. Instead, it might be more helpful to notice the sea change in the industry that has resulted from the one-two punch of the SDK and the App Store. Carriers that had locked down their networks are now racing to open app stores of their own, terrified that every cool new app for the iPhone potentially lures away more of their customers. Some mobile developers are going iPhone-only. Given the realization that with iPhone users far more likely to download and pay for applications, it's a sucker's bet to work through the miseries of other mobile platforms—the aforementioned fragmentation and obstructionist carriers—in the hopes of reaching a user base that doesn't download third-party apps anyway. And in the enterprise, companies are writing their own custom iPhone applications and deploying them en masse in the field.

If you're reading this book, chances are you'd like to get in on this mobile revolution.

And we're glad you do, because we're eager to help see you through the journey that will take you from downloading the SDK to releasing your first application.

We begin by assuming only two things:

- You have a computer capable of running the current iPhone SDK. As of this writing, that means a Mac with an Intel CPU, running Mac OS X Leopard 10.5.4 or later.¹
- You are familiar with a “curly brace” programming language (C, C++, C#, Java, Ruby, etc.) and with object-oriented programming in general. If you’re primarily familiar with scripting languages (JavaScript, ActionScript, PHP), then you may find some of the iPhone’s programming practices challenging, but we hope not insurmountable.

1.1 In This Book

We begin your journey with an introduction to the platform. We start with a quick success in the form of a Hello World iPhone application. Fresh on the heels of your first success, we dive into some of the fundamentals of the platform that you will need going forward. We cover the basics of the tools (Xcode, Interface Builder) and Objective-C, which is the language of Cocoa Touch development.

In the next four chapters, we look at view controllers and the views that are integral to most iPhone applications. You’ll begin with a look at the most generic views and view controllers and move to the UI metaphors that are key to almost any iPhone application: tab bars, navigation, and table views.

We recommend that you read these chapters no matter what sort of iPhone application you are planning on building. You can then work through the rest of the book in order, or you can pick and choose according to your needs. We’ve grouped the remaining chapters into roughly six sections.

If you need to persist data to your device, you’ll find four chapters that cover the various techniques. We start with filesystem access and options for saving and retrieving user preferences. Then we show two approaches to using the built-in database: directly, with the SQLite API,

1. SDK requirements, particularly the minimum OS version, are highly likely to change going forward. Check <http://developer.apple.com/iphone> for the current specs.

or via the object-relational mapping power of the Core Data framework added in iPhone OS 3.0.

Your application may require communication with servers or other devices, so the next two chapters take you out to the network. The first connects to websites, web services, and email on the public Internet, while the second uses the self-networking abilities of Bonjour and the new-in-3.0 Game Kit framework to connect iPhone OS devices to one another.

The next section consists of four chapters on interacting with media on your device. We start with the simple framework for viewing video on the iPhone and then look at 3.0's new framework for interacting with the user's iPod music library. For developers who want to exploit the platform's audio capabilities, we have a chapter on playing and recording audio with 3.0's new AVFoundation framework, closing the section with an overview of the lower-level Core Audio framework.

You are probably going to want to customize the way your users interact with your application. We return to presentation with a look at events, Multi-Touch, and gestures. Custom drawing via custom views is the topic of discussion in the next three chapters. We discuss drawing content in custom and unique ways with the sophisticated Core Graphics library, as well as manipulating and drawing images, both those provided by your application as well as those from the user's image library. This section closes with a chapter on Core Animation, the technology that underlies all the beautiful and natural-feeling animations that you see throughout the iPhone UI.

Next, we cover the more specialized technologies that not every iPhone application will use but that, when used properly, really make your application stand out. Some developers will be interested in using the built-in accelerometer to sense the device's orientation and motion, allowing the user to control applications in new and interesting ways. Developers of productivity and social networking applications will be able to use the Address Book framework to interact with a user's contact database. And because the iPhone and iPod touch are devices on the move, we have two chapters on location: one on the Core Location framework that helps determine the device's location and a second on the new-in-3.0 Map Kit framework that lets you present map data in your application.

Our final section offers four chapters on the final steps you'll need to take to complete your application. A handful of “application integration” APIs allow your application to launch other apps and to be launched by others. You'll learn to perfect your application with chapters on debugging your code and improving its performance with Xcode's various tools. A final chapter puts it all together by helping you hone your development process, get your work onto the App Store, and promote it to the app-shopping public.

1.2 Acknowledgments

From Chris Adamson

When I was part of an office-emptying mass layoff in 2001, I swore that I was done with mobile application development. A few years later, crawling to the finish of my second book, I said I'd never write another one. So, to be writing an introduction to a book on iPhone development, it would seem I've got some explaining to do.

It really all comes down to one moment in March 2008. I was at a Java conference, skeptically following the web coverage of the iPhone SDK announcement. Few people, myself included, thought we'd get a substantial level of functionality in the SDK and were shocked when what was unveiled was most of Mac OS X's Cocoa, with the UI rethought for a touch-based handheld. While I'd never been able to get out to the network with carrier-crippled Java phones, the SDK offered *freakin' BSD sockets* and self-networking with Bonjour as a bonus. Media, often presented at a toy level of functionality on other platforms, was present in the form of Core Audio, the same code used by professional audio applications on the Mac.

And this was all for a phone? I was hooked. I call it my “All Along the Watchtower” moment: I immediately knew that everything I'd been doing for years was instantly rendered irrelevant and that this is what I wanted—needed—to be doing going forward. I got the SDK over a very slow DSL connection that night, coded my first Hello World on the flight home, and was recruited for this book a few months later.

Of course, I have to thank our editor Daniel Steinberg for reaching out to me and convincing me that I was a good choice to coauthor the book. He's kept this enormous project on track through months of writing and rewriting, a shutdown when we feared Apple's nondisclosure agreement might keep us from ever releasing it, and the first slam-it-together

beta release when the NDA dropped. Prags Andy and Dave also deserve kudos for a very productive (and yes, I'll say it, *agile*) system for book writing. Bill has been a responsive and productive coauthor, someone whose strengths are a well-suited balance to my weaknesses. Both of us are grateful to our many technical reviewers and the huge community that has developed on <http://pragprog.com/>, in the book's forum and errata page, giving us tons of useful feedback. And thanks as always to my family (Kelly, Keagan, and Quinn), who somehow withstood the collateral damage and stress of not only this book-writing marathon but also a cross-country move just a couple weeks before beta 1.

Obligatory end-of-book tune check: this time it was Bend Sinister, My Chemical Romance, Rilo Kiley, The Polyphonic Spree, Immaculate Machine, . . . And You Will Know Us by the Trail of Dead, The Cribs, Kaki King, and the CBC Radio 3 stream and podcasts.²

From Bill Dudney

Writing a book is a big task. But of course it's not just the authors who work hard to produce the content. The editors (at least the ones who have corrected my prose) work just as hard if not harder to turn the technobabble and passive writing into acceptable English, so thanks to Daniel for helping me over the hump again.

The iPhone engineering team of course deserves a huge thanks for working nights and weekends to make something that is so much fun to build for.

The reviewers were also very generous with their time. Many helped, but a few stick in my mind as being especially helpful, so in no particular order let me say a hardy thanks to the following people: Tim Isted, Steven Troughton Smith, Patrick Burleson, Jason Hunter, Andy Simmons, Marcus Crafter, Tom Hauburger, David Hodge, David Smith, and Dee Wu. The book is much better for all your hard work!

I'd also like to thank the many folks who have been in the iPhone Studio for asking just the right question to help me see things from a different perspective. Those questions and the shift it caused helped to make this book. Thanks!

My family was also great during the long journey that has been this book. Thanks to you all for allowing me to stay up late nights and work

2. You can find up-to-date stats at <http://www.last.fm/user/invalidname>.

early mornings (and most of the time in between) to finish this up. I'd especially like to thank my oldest son, Andrew, who has been a great help in finding the things that I'm constantly forgetting. And finally, I'd like to thank the 2,000-year-old Jewish carpenter for making my life more than I ever thought it would or could be.

We are really excited about this platform and what you are going to build with your newfound knowledge. It is a great time to be an iPhone developer, and it's great to have you as part of the community!

The Pragmatic Bookshelf

The Pragmatic Bookshelf features books written by developers for developers. The titles continue the well-known Pragmatic Programmer style and continue to garner awards and rave reviews. As development gets more and more difficult, the Pragmatic Programmers will be there with more titles and products to help you stay on top of your game.

Visit Us Online

iPhone SDK Development's Home Page

<http://pragprog.com/titles/amiphd>

Source code from this book, errata, and other resources. Come give us feedback, too!

Register for Updates

<http://pragprog.com/updates>

Be notified when updates and new books become available.

Join the Community

<http://pragprog.com/community>

Read our weblogs, join our online discussions, participate in our mailing list, interact with our wiki, and benefit from the experience of other Pragmatic Programmers.

New and Noteworthy

<http://pragprog.com/news>

Check out the latest pragmatic developments, new titles and other offerings.

Buy the Book

If you liked this eBook, perhaps you'd like to have a paper copy of the book. It's available for purchase at our store: pragprog.com/titles/amiphd.

Contact Us

Online Orders:	www.pragprog.com/catalog
Customer Service:	support@pragprog.com
Non-English Versions:	translations@pragprog.com
Pragmatic Teaching:	academic@pragprog.com
Author Proposals:	proposals@pragprog.com
Contact us:	1-800-699-PROG (+1 919 847 3884)