

*Making Everything Easier!™*

# iOS App Development

FOR  
**DUMMIES**<sup>®</sup>  
A Wiley Brand

## **Learn to:**

- Download the iOS SDK and use Apple's developer tools
- Build a universal app for the iPad<sup>®</sup> and iPhone<sup>®</sup>
- Make the most of the latest iOS features in your app designs
- Provide a great user experience and make your app stand out in the crowd

**Jesse Feiler**

*Author of iWork For Dummies<sup>®</sup>, 2<sup>nd</sup> Edition*





**by Jesse Feiler**



## **iOS App Development For Dummies®**

Published by: **John Wiley & Sons, Inc.**, 111 River Street, Hoboken, NJ 07030-5774, [www.wiley.com](http://www.wiley.com)

Copyright © 2014 by John Wiley & Sons, Inc., Hoboken, New Jersey

Published simultaneously in Canada

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 the prior written permission of the Publisher. 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>.

**Trademarks:** Wiley, For Dummies, the Dummies Man logo, Dummies.com, Making Everything Easier, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

<p><b>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</b></p>
---

**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, 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. For technical support, please visit [www.wiley.com/techsupport](http://www.wiley.com/techsupport).

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included

in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit [www.wiley.com](http://www.wiley.com).

Library of Congress Control Number: 2013957974

ISBN 978-1-118-87105-8 (pbk); ISBN 978-1-118-87107-2 (ebk); ISBN 978-1-118-87110-2 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

# iOS App Development For Dummies®

Visit

[www.dummies.com/cheatsheet/iosappdevelopment](http://www.dummies.com/cheatsheet/iosappdevelopment) to view this book's cheat sheet.

## Table of Contents

### Introduction

#### A Bit of History

[The iPhone stands alone](#)

[Enter the App Store](#)

[The iPad joins the party](#)

#### The Plan for This Book

#### iOS and Xcode Game Changers

#### About This Book

#### Conventions Used in This Book

#### Icons Used in This Book

#### Foolish Assumptions

#### How This Book Is Organized

[Part I: Getting Started](#)

[Part II: Building RoadTrip](#)

[Part III: Getting Your Feet Wet: Basic Functionality](#)

[Part IV: The Model and the App Structure](#)

[Part V: Adding the App Content](#)

[Part VI: The Part of Tens](#)

#### Beyond the Book

#### Where to Go from Here

### **Part I: Getting Started**

#### **Chapter 1: What Makes a Great iOS App**

## [Figuring Out What Makes a Great iOS App](#)

[Making your app work well](#)

[Handling networking, social media, and location](#)

[Designing a powerful and intuitive interface that disappears](#)

## [Using the iOS Platform to the Fullest](#)

[Exploiting advantages of the system](#)

[Accessing the Internet](#)

[Knowing the location of the user](#)

[Tracking orientation and motion](#)

[Tracking users' fingers on the screen](#)

[Playing content](#)

[Accessing information from Apple's apps](#)

[Copying, cutting, and pasting between apps](#)

[Multitasking, background processing, and notifications](#)

[Living large on the big screen](#)

## [Embracing Device Limitations](#)

[Designing for fingers](#)

[Balancing memory and battery life](#)

## [Why Develop iOS Apps?](#)

### [Developing with Apple's Expectations in Mind](#)

### [Thinking About You, Apps, and Money](#)

### [Enter the Cloud](#)

### [Developing an App the Right Way Using the Example App in This Book](#)

### [What's Next](#)

## **[Chapter 2: Getting to Know the SDK](#)**

### [Developing Using the SDK](#)

#### [Using Xcode to Develop an App](#)

[Creating an Xcode project](#)

[Developing the app](#)

#### [The Workspace Window](#)

[Workspace areas](#)

[Displaying an area's content](#)

[The toolbar and Tab bar](#)

[The Organizer window](#)

## **[Chapter 3: The Nuts and Bolts of an Xcode Project](#)**

### [Creating Your Project](#)

#### [Exploring Your Project](#)

[The project](#)

[The Project editor](#)

[The Project navigator](#)

#### [Setting Your Xcode Preferences](#)

#### [Building and Running Your Application](#)

[Building an app](#)

[The iPad's Split views](#)

[The Log navigator](#)

## [Running in the Simulator](#)

[Interacting with your simulated hardware](#)

[Making gestures](#)

[Uninstalling apps and resetting your device](#)

[Living with the Simulator's limitations](#)

## [Using Asset Catalogs](#)

## [Adding the Image and Sound Resources and an App Icon](#)

# **[Part II: Building RoadTrip](#)**

## **[Chapter 4: Storyboards and the User Experience](#)**

### [Introducing the Storyboard](#)

[Telling your story](#)

[Working with object graphs](#)

### [Defining What You Want an App to Do: The RoadTrip App](#)

### [Creating the Application Architecture](#)

### [What You Add Where](#)

### [Using Frameworks](#)

### [Using Design Patterns](#)

[The iOS design patterns](#)

[The Model-View-Controller \(MVC\) design pattern](#)

### [Working with Windows and Views](#)

[Looking out the window](#)

[Admiring the view](#)

[The kinds of views you use](#)

### [View Controllers — the Main Storyboard Players](#)

### [What About the Model?](#)

### [It's Not That Neat](#)

### [Taking a Look at Other Frameworks](#)

[The Foundation framework](#)

[The CoreGraphics framework](#)

[Even more frameworks](#)

### [Understanding the MVC in the Project](#)

## **[Chapter 5: Creating the RoadTrip User Interface](#)**

### [Creating Your User Interface in the iPad Storyboard](#)

[It's about the view controller](#)

[Using Interface Builder to add the user elements](#)

### [Working within the Utility Area](#)

[Inspector and Quick Help pane](#)

[Library pane](#)

### [Understanding iPad Navigation](#)

### [Adding a New View Controller](#)

[Danger Will Robinson](#)

[Adding an identifier to the view controller](#)

### [View Layout](#)

### [Adding the User Interface Objects](#)

[Autosizing with Auto Layout](#)

[Adding the Test Drive button](#)

[Massaging the Template Code](#)

[Getting Rid of Warnings](#)

[Creating the iPhone User Interface](#)

## **Chapter 6: The Runtime, Managing Memory, and Using Properties**

[Stepping Through the App Life Cycle](#)

[UIApplicationMain](#)

[Handling events while your application is executing](#)

[Knowing what to do when the normal processing of your application is interrupted](#)

[An overview of the view controller life cycle](#)

[Working within the Managed Memory Model Design Pattern](#)

[Understanding memory management](#)

[Using reference counting](#)

[Automatic Reference Counting \(ARC\)](#)

[Working with variable types according to ARC](#)

[Understanding the deadly retain cycle](#)

[Observing Low-Memory Warnings](#)

[The didReceiveMemoryWarning method](#)

[The applicationDidReceiveMemoryWarning: method](#)

[The UIApplicationDidReceiveMemoryWarningNotification: notification](#)

[Picking the right memory-management strategy for your application](#)

[Customizing the Behavior of Framework Classes](#)

[Subclassing](#)

[The Delegation pattern](#)

[Understanding Declared Properties](#)

[What comprises a declared property](#)

[Using dot syntax](#)

[Setting attributes for a declared property](#)

[Writing your own accessors](#)

[Accessing instance variables with accessors](#)

[Hiding Instance Variables](#)

## **Chapter 7: Working with the Source Editor**

[Navigating in the Xcode Source Editors](#)

[Using the Jump bar](#)

[Organizing your code using the #pragma mark statement](#)

[Using the Xcode Source Editor](#)

[Using Live Issues and Fix-it](#)

[Compiler warnings](#)

[The Issue navigator](#)

[Accessing Documentation](#)

[Getting Xcode help](#)

[The Organizer window](#)

[The Help menu](#)

[Finding and Searching in Your Project](#)

[Using the Find command to locate an item in a file](#)

[Using the Search navigator to search your project or framework](#)

[Using the Symbol navigator](#)

[You're Finally Ready to Write Code!](#)

## **Part III: Getting Your Feet Wet: Basic Functionality**

### **Chapter 8: It's (Finally) Time to Code**

[Checking for Network Availability](#)

[Downloading the Reachability sample](#)

[Adding the code to check for reachability](#)

[Exploring the Changes in iOS 7](#)

[The dated interface](#)

[Losing the content](#)

[Setting the Master View Controller Title](#)

[Understanding Autorotation](#)

[Writing Bug-Free Code](#)

[Working in the Debug area and Debug navigator](#)

[Managing breakpoints](#)

[What you'll find in the Debug area](#)

[What you'll find in the Debug navigator](#)

[Displaying variables in the Source editor](#)

[Tiptoeing through your program](#)

### **Chapter 9: Adding Outlets and Actions to Your RoadTrip Code**

[Using Custom View Controllers](#)

[Adding the custom view controller](#)

[Setting up the TestDriveController in the MainStoryboard for iPad](#)

[Understanding Outlets](#)

[Adding Outlets](#)

[Opening the Assistant editor](#)

[Creating the outlet](#)

[The Connections inspector](#)

[Working with the Target-Action Design Pattern](#)

[Using the Target-Action pattern: It's about controls](#)

[Adding an action](#)

[How Outlets and Actions Work](#)

[Update the iPhone storyboard file](#)

### **Chapter 10: Adding Animation and Sound to Your App**

[Understanding iOS Animation](#)

[View geometry and coordinate systems](#)

[Points versus pixels](#)

[A view's size and position](#)

[Working with data structures](#)

[Coordinating Auto Layout, Frames, and Constraints](#)

[Animating a View](#)

[Finally, More Code](#)  
[Implementing the testDrive Method](#)  
[Understanding Block Objects](#)  
[Rotating the Object](#)  
[Working with Audio](#)  
[Tracking Touches](#)  
[Animating a Series of Images “In Place”](#)  
[iPhone versus iPad](#)

## **Part IV: The Model and the App Structure**

### **Chapter 11: The Trip Model**

[What the Model Contains](#)  
[Adding the Model Data](#)  
[Using property lists](#)  
[Adding a property list to your project](#)  
[Adding the First Two Model Classes](#)  
[Understanding the Trip Interface](#)  
[Implementing the Trip Class](#)  
[Initializing objects](#)  
[Invoking the superclass’s init method](#)  
[Initializing instance variables](#)  
[Returning self](#)  
[Initializing the Destination Class](#)  
[Creating the Trip Object](#)  
[More Debugger Stuff](#)

### **Chapter 12: Implementing the Master View Controller**

[Setting Up a Custom View Controller for the iPad](#)  
[Adding a Background Image and Title](#)  
[Updating the iPhone Storyboard File](#)

### **Chapter 13: Working with Split View Controllers and the Master View**

[The Problem with Using a Navigation Controller in Detail View](#)  
[Using a Navigation Controller in the Master View](#)  
[Adding a Gesture Recognizer](#)  
[The Split View Controller](#)  
[The UISplitViewController delegate](#)  
[Localization](#)  
[Back to the main feature](#)  
[Adding the Toolbar](#)  
[Adding the button when the view controller is replaced](#)  
[A Few More Tweaks to the MasterViewController](#)  
[And \(a Little Extra\) One More Thing](#)

[Don't Forget the iPhone](#)

## **Chapter 14: Finishing the Basic App Structure**

[Extending the iPad Storyboard to Add More Functionality to Your App](#)

[Adding the Weather view controller](#)

[Adding the Events controller](#)

[Adding the remaining controllers](#)

[Changing the Split View Controller to a Detail View Controller Relationship](#)

[Repeat for iPhone](#)

## **Part V: Adding the App Content**

### **Chapter 15: How's the Weather? Working with Web Views**

[The Plan](#)

[The iPad storyboard](#)

[The iPhone storyboard](#)

[Setting Up the Weather Controller](#)

[Adding the custom view controller](#)

[Setting Up WeatherController in the Main iPad.storyboard file](#)

[The Weather Controller](#)

[Managing links in a Web view](#)

[More Opportunities to Use the Debugger](#)

[Unrecognized selector sent to instance](#)

[Repeat for the iPhone Storyboard](#)

[Adding the WeatherController to the iPhone storyboard file](#)

[Test in the iPhone Simulator](#)

### **Chapter 16: Displaying Events Using a Page View Controller**

[The Plan](#)

[Setting Up the EventsController](#)

[Adding the custom view controller](#)

[Setting up the EventsController in the MainStoryboard](#)

[Adding and setting up the EventPageController in the MainStoryboard](#)

[Extending the Trip Model](#)

[Adding the Events Class](#)

[The EventsController and Its PageViewController](#)

[Data sources and delegates](#)

[Data source](#)

[Delegate](#)

[The EventsController](#)

[The EventPageController](#)

[Adding Events Support to the iPhone Storyboard](#)

### **Chapter 17: Finding Your Way**

[The Plan](#)

[Setting Up the Map Controller](#)

[Adding the custom view controller](#)

[Setting up the MapController in the Main iPad.Storyboard](#)

[Test](#)

## [Putting MapKit through Its Paces](#)

[MKMapView](#)

[Enhancing the map](#)

## [Changing the Map Type](#)

### [Adding Annotations](#)

[Creating the annotation](#)

[Displaying the map title and annotations](#)

### [Going to the Current Location](#)

### [Fixing the Status Bar](#)

### [Update the iPhone Storyboard](#)

## **[Chapter 18: Geocoding](#)**

[Understanding Geocoding on the iPad](#)

[Reverse Geocoding](#)

## **[Chapter 19: Finding a Location](#)**

### [Setting Up the Find Controller](#)

[Adding the custom view controller](#)

[Setting up FindController in the Main iPad File](#)

### [Implementing the Find Controller](#)

[Adding the Map View](#)

[Getting the text](#)

[Disabling cell highlighting](#)

### [Finding the Location](#)

[Making the Map Title the Found Location](#)

[Adding the FindController to the iPhone Storyboard](#)

## **[Chapter 20: Selecting a Destination](#)**

### [The Plan](#)

### [Setting Up the DestinationController for the iPad Storyboard](#)

[Adding the custom view controller](#)

[Setting up the DestinationController in the Main iPad.storyboard](#)

### [Adding a Modal View](#)

### [Implementing a Table View](#)

### [Creating the Table View](#)

[Adding sections](#)

[Displaying the cell](#)

[Working with user selections](#)

### [Saving the Destination Choice and Selecting a Destination](#)

[Displaying the Destination table](#)

[Testing](#)

### [Adding Destination Support to the iPhone Storyboard](#)

### [A Word about Adding Settings](#)

### [What's Next?](#)

## **Part VI: The Part of Tens**

### **Chapter 21: Ten Ways to Be Successful with Apps**

[Make a Million Dollars in a Week](#)

[Build a Portfolio](#)

[Build App Icons](#)

[Design User Interfaces](#)

[Build Back Ends](#)

[Socialize with Apps](#)

[Talk About Apps with People Who Want Them](#)

[Promote Apps](#)

[Provide Support to Users](#)

[Fix Bugs](#)

### **Chapter 22: Ten Ways to Be a Happy Developer**

[Keep Things Loosely Coupled](#)

[Remember Memory](#)

[Don't Reinvent the Wheel](#)

[Understand State Transitions](#)

[Do the Right Thing at the Right Time](#)

[Avoid Mistakes in Error Handling](#)

[Use Storyboards](#)

[Remember the User](#)

[Keep in Mind That the Software Isn't Finished Until the Last User Is Dead](#)

[Keep It Fun](#)

**About the Author**

**Cheat Sheet**

**More Dummies Products**

# Introduction

---

*iOS App Development For Dummies* is a beginner's guide to developing iOS apps. And not only do you *not* need any iOS development experience to get started, but you also don't need any Mac development experience, either. I've written this book as though you are coming to iPhone and iPad app development as a blank slate, ready to be filled with useful information and new ways to do things. Well, almost a blank slate, anyway; see the upcoming "[Foolish Assumptions](#)" section for details on what you *do* need to know before using this book.

Because of the nature of the iPhone and iPad, you can create small, bite-sized apps that can be quite powerful. Also, because you can start small and create real applications that do something important for a user, it's relatively easy to transform yourself from an "I know nothing" person into a developer who, though not (yet) a superstar, can still crank out quite a respectable app.

But the iPhone and iPad can be home to some pretty fancy software as well — so I'll take you on a journey through building an industrial-strength app and show you the ropes for developing one on your own.

## ***A Bit of History***

It's 6:00 a.m. PST on January 9, 2007. A distressingly long line of nerds wrapped all the way around San Francisco's Moscone Center. Why? To hear Steve Jobs give his MacWorld Expo keynote address. It was nuts to get up so early on a cold morning, but Steve Jobs was rumored to be introducing an Apple phone.

No one knew whether an Apple phone would be any good, but perhaps Steve would show us magic — something that would revolutionize an industry. Perhaps it would be as cool and important as the iPod! A few hours later, Steve told the crowd that “Apple is going to reinvent the phone.” Steve was never modest, but he was certainly correct — Apple completely blew away the phone industry that day. The damage was not yet visible to the current phone vendors (Palm, Motorola, Nokia, Sony, Ericsson, RIM, and Microsoft), but they were suddenly left back in the 20th century. The future had arrived.

## ***The iPhone stands alone***

The first iPhone shipped in late June 2007. It came with a bunch of Apple’s native apps such as Contacts, Maps, Stocks, Safari, and so on. The problem was that only Apple could develop these native apps. The Apple developer “evangelists” told developers that we should be happy writing web apps. This did not make us happy — we wanted to write native Objective-C apps.

Steve Jobs and Apple eventually saw the light (in fact, some people believe that there was always the possibility of releasing tools to let developers write native apps, but getting the iPhone itself launched took a higher priority). Apple released a beta version of the iPhone Software Development Kit (SDK) in the spring of 2008, and it opened the App Store for business in July 2008. At this point, you could develop native apps — but only for the iPhone, because the iPad did not yet exist.

## ***Enter the App Store***

The App Store in July 2008 was a far cry from today’s App Store. The numbers of apps and the numbers of downloads today are staggering. Search for “App Store”

on Wikipedia to get the latest numbers. The billions of dollars that developers have earned directly from the App Store are fantastic.

But beyond these large numbers, there's something about the App Store that I didn't truly appreciate until my first app went on sale. On the first day, more than 20 copies were sold. (My initial advertising was a mass e-mail to friends.) This was very much a niche-of-a-niche product, but it continued to sell a few copies each week. I added a link to the app to my e-mail signature, and, when I saw a few copies had been sold in Great Britain, I assumed that some of my English cousins had pitched in (bless their hearts, as they say in the South).

But I don't have any relatives in Argentina. I'm pretty certain I don't know anyone in Malaysia. Okay, the first couple of sales in Canada might be explained by the fact that I live 20 miles south of the border. But why would someone in China be buying the app? I certainly hope the people in South Africa who have bought the app are using it productively.

Almost all of those people found the app by searching on the App Store. Apple provides a great deal of help and advice for you to put your app's best face forward on the App Store, and they provide tips and continually refine their search algorithms so that if you use good keywords, people can discover your apps. Apple wants to sell hardware, and they want users of their devices to discover apps that enhance their experiences with the devices.

The numbers of iOS devices are so vast that, with good keywords and a good app description, a niche-of-a-niche-of-a-niche app can find a home on the App Store. You may write the next blockbuster app, but you also may write an app that gets modest results. The highly

automated App Store provides the infrastructure to make it all possible.

You can count me among the people who think that the App Store itself may turn out to be a more significant achievement for Apple than the iPhone itself.

## ***The iPad joins the party***

Apple released the first iPad in April 2010. In some ways, the iPad was an even more remarkable achievement than the iPhone. The mobile phone existed before iPhone. iPad was the first time that a high-powered computing and communicating device that was truly mobile caught on.

Initially, the iPad ran the iPhone OS. That was a little hard for some people to understand, and before long, the operating system was renamed iOS. We're now at version 7 of iOS. In addition to the renaming, there has been some restructuring of the developer tools and environment so that things fit together very well.

When I look at developer features such as universal apps (they can run on both iPhone and iPad with minimal code changes), support for in-app purchases, and iBeacon integration, I see a full-featured environment that matches and even surpasses some of the most sophisticated development environments I've worked with.

Even though there are many more features today than there were back in 2008, iOS development today is easier than it was a few years ago. The developer tools have matured, and the frameworks themselves have been tweaked with tools such as auto layout that make the placement of interface elements on a screen automatic as devices are rotated and as new screen sizes appear on the devices.

# *The Plan for This Book*

You will build this book's RoadTrip app using the following steps:

1. Create the initial storyboards for both the iPad and iPhone versions, starting with Xcode's Master-Detail project template.

The template's iPad storyboard is based on using UIKit's `UISplitViewController`, which uses the same custom `MasterViewController` and custom `DetailViewController` used in the iPhone version. The Master View controller will appear on the left when the iPad is held in landscape orientation, whereas the Detail View controller appears on the right.

2. Build and test the iPad version in the iPad simulator. You should see a Table view embedded in a Navigation controller in the Master view.

The template's initial iPhone storyboard design begins with a custom `MasterViewController` (a Table view) embedded in a Navigation view. Selecting an item in the Table view displays data managed by a custom `DetailViewController`.

3. Build and test the iPhone version in the iPhone simulator. It should also work perfectly because you haven't had a chance to make any mistakes yet.
4. Add a `TestDriveController` to the iPhone storyboard. Build and test. Add it to the iPad storyboard. Build and test.
5. Add animation and sound to the Test Drive controller. Build and test both the iPhone and iPad versions.

6. Add additional features to each version until done.

## *iOS and Xcode Game Changers*

With iOS 7 and — more importantly — with Xcode 5 (and later versions), the nuts and bolts of iOS app development have changed dramatically. Xcode 5 has added much more functionality to the integrated development environment (IDE) you use to develop iOS applications, especially when it comes to writing syntactically correct (and bug-free) code that's better able to manage memory. The latest versions bring much simpler integration with the App Store as well as newly designed performance-monitoring tools. Of course, the rub is getting the hang of Xcode 5. That's where this book comes in. I carefully take you through Xcode 5, pointing out its features and describing how to best use them. When you set this book aside, you'll have a great understanding of how to take advantage of all those features that will make your life easier.

You find out how to develop a single app that includes features that readers and students of earlier editions have been asking for — including more animation and sound — as well as an infrastructure that people can use to develop more robust applications. The resulting example is an app called RoadTrip, which can send you on your way to developing apps that you can be proud of and that other people will want to have.

This new edition is based on iOS 7 *and* Xcode 5. If you want to find out how to develop applications, the tools discussed in this book are the tools you absolutely need to use to do it the right way.

# *About This Book*

*iOS App Development For Dummies* distills the hundreds (or even thousands) of pages of Apple documentation (not to mention my own development experience and that of many colleagues and friends) into only what's necessary to start you developing real applications. But this is no recipe book that leaves it up to you to put it all together. Rather, it takes you through the frameworks (the code supplied in the SDK) and iOS architecture in a way that gives you a solid foundation in how applications really work, and also acts as a road map to expand your knowledge as you need to.

I assume that you're in this for the long haul and that you want to master the whole app-development ball of wax. I use real-world applications to show the concepts and give you the background on how things actually work on iOS.

For many people, their first excursion into programming has been the classic Hello World C program. Depending on how you space it, it can be written in anywhere from one to three lines of code. For a long time now, I've thought that this program has long since outlived its usefulness. We just don't write code like that today (and many of us never did). As you'll see in this book, the development process involves typing code, but it also involves drawing an interface with your mouse, choosing options in check boxes to configure your app, and a whole bunch of other activities that are as far as you can get from typing a few lines of code into a blank document.

It's a new world for developers, and I think its excitement and opportunities have barely started.

# *Conventions Used in This Book*

This book guides you through the process of building iOS apps. Throughout the book, you use the classes provided by Apple’s iOS frameworks (and create new classes of your own, of course). You code them using the Objective-C programming language.

Code examples in this book appear in a monospaced font so that they stand out a bit better. That means that the code you see will look like this:

```
#import <UIKit/ UIKit.h>
```

Objective-C is based on C, which *is* case-sensitive, so please enter the code that appears in this book *exactly* as it appears in the text. I also use the standard Objective-C naming conventions — for example, class names always start with a capital letter, and the names of methods and instance variables always start with a lowercase letter.

Note that all URLs in this book appear in a monospaced font as well, like this:

```
www.northcountryconsulting.com
```

When I ask you to add code to a program, it will be in bold like this:

```
#import <UIKit/ UIKit.h>
```

You’ll notice — starting around Chapter [8](#) — that you will sometimes be asked to delete some of the code you have in place for your project to make room for some new stuff. When that happens, I comment out the code to make things really clear. I refer to code I want you to delete as commented out, bold, underline, and italic code

because said code will show up as commented out, bold, underlined, and italic. Simple enough, as shown in the following example:

```
// Delete this
```

If you're ever uncertain about anything in the code, you can always look at the source code on the companion website. (More on that in the section "[Beyond the Book](#)," later in this Introduction.)

## *Icons Used in This Book*



This icon indicates a useful pointer that you shouldn't skip.



This icon represents a friendly reminder. It describes a vital point that you should keep in mind while proceeding through a particular section of the chapter.



This icon signifies that the accompanying explanation may be informative (dare I say interesting?), but it isn't essential to understanding iOS app development. Feel free to skip past these tidbits if you like (though skipping while learning may be tricky).



This icon alerts you to potential problems that you may encounter along the way. Read and obey these blurbs to avoid trouble.



This icon indicates how to use an important part of Xcode functionality. This helps you wade through Xcode's complexity and focus on how to get specific things done.

## ***Foolish Assumptions***

To begin programming your iOS applications, you need an Intel-based Mac with the latest or next-to-latest version of OS X on it. (No, you can't develop iOS applications directly on the iPhone or iPad.) You also need to download the iOS Software Development Kit (SDK) — which is free. And, oh yeah, you need an iPhone and/or iPad. You won't start running your app on it right away — you'll use the iOS Simulator that Apple provides with the iOS SDK during the initial stages of development — but at some point, you'll want to test your app on a real, live iOS device.

This book assumes that you have some programming knowledge and that you have at least a passing acquaintance with object-oriented programming, using some variant of the C language (such as C++, C#, or maybe even Objective-C). In case you don't, I'll point out some resources that can help you get up to speed. The app example in this book is based on the frameworks that come with the SDK; the code is pretty simple (usually) and straightforward. (I don't use this book as a platform to dazzle you with fancy coding techniques.)

I also assume that you're familiar with the iPhone and iPad themselves and that you've at least explored Apple's included applications to get a good working sense of an iOS app's look and feel. It might also help to browse the App Store to see the kinds of applications available there

and maybe even download a few free ones (as if I could stop you).

# ***How This Book Is Organized***

*iOS App Development For Dummies* has six main parts, which are described in the following sections.

## ***Part I: Getting Started***

Part [I](#) introduces you to the iOS world. You find out what makes a great iOS app and see how an iOS app is structured. In Chapter [2](#), I give an overview of how Xcode 5 works that gets you up to speed on all its features; you can use this chapter as a reference and return to it as needed. You also create your Xcode project in this part — a universal app that can run equally well on an iPad or iPhone — and I take you on a guided tour of what makes up the Xcode project that will become your home away from home.

## ***Part II: Building RoadTrip***

In this part of the book, you find out how to create the kind of user interface that will capture someone's imagination. I explain the Interface Builder editor, which is much more than your run-of-the-mill program for building graphical user interfaces. You also discover storyboards, which are the icing on the Interface Builder cake that let you lay out the entire user experience and app flow — saving you a lot of coding, to boot.

You'll also take a brief tour of the RoadTrip app, the app that you build in this book. I show you not only what the app can do but also how it uses the frameworks and SDK to do that.

I also explain how the main components of an iOS app go together. I describe how the iOS applications work from

a viewpoint of classes and design patterns, as well as show how the app works at runtime. I spend some time on three very important ideas: how to extend the framework classes to do what you want them to, how to manage memory, and how to take advantage of declared properties. I also explain how everything works together at runtime, which should give you a real feel for how an iOS app works.

Parts [I](#) and [II](#) give you the fundamental background that you need to develop iOS applications.

### ***Part III: Getting Your Feet Wet: Basic Functionality***

Now that you have the foundation in place, Part [III](#) starts you on the process of having your app actually *do* something. You start off by determining whether a network is available to support the app functionality that requires Internet access. You find out how to customize the appearance of the controls provided by the framework to make your app a thing of beauty. You finish off by adding animation and sound just to get going. You also see how to connect the elements on your storyboard to your app code to make them do things — such as have a '59 pink Cadillac Eldorado Biarritz convertible drive up and down the screen.

### ***Part IV: The Model and the App Structure***

Now you begin to get down to the real work. You find out about the iPad's popovers and Split View controllers, and you also add navigation to the app. Along the way, I really get into showing you how to account for the differences between an iPad and an iPhone, and make sure that the app can run flawlessly on whatever device the user has handy. You also add the app model, which

provides both the data and the logic you need to create an app that delivers real value to the user. You then finish the storyboard so that you can see your basic application flow. To wrap it all up, I show you how to package your app with a custom icon and prepare it for the App Store.

## ***Part V: Adding the App Content***

Now that you have the application foundation and the user experience architecture in place, Part [V](#) takes you into the world of applications that contain major functionality. I show you how to display the weather using a web page right off the Internet, how to allow the user to page through local events as if he were reading a book, how to display a map of where the user is going and where he is right now, how to find a location that he has always wanted to visit and display it on a map, and even how to change where he is going (limited in the RoadTrip app to New York and San Francisco, but it's incredibly easy to add other destinations). I don't go slogging through every detail of every detail, but I demonstrate almost all the technology you need to master if you intend to create a compelling app like this on your own.

## ***Part VI: The Part of Tens***

Part [VI](#) consists of some tips to help you avoid having to discover everything the hard way. It talks about approaching app development in an “adult” way right from the beginning (without taking the fun out of it). I also revisit the app and explain what else you would need to do to make this app a commercial and critical success.

# ***Beyond the Book***

This book has additional content you can access online:

- ✓ **Sample code:** Sample code for each chapter can be found online at [www.dummies.com/extras/iosappdevelopment](http://www.dummies.com/extras/iosappdevelopment). Note that the posted code represents the code as it is at the end of the chapter. If you want to download code to follow along with as you read, download the code for the previous chapter.
- ✓ **Cheat Sheet:** This book's Cheat Sheet can be found online at [www.dummies.com/cheatsheet/iosappdevelopment](http://www.dummies.com/cheatsheet/iosappdevelopment). See the Cheat Sheet for more on expanding your app with subclassing, target action, and delegation.
- ✓ **Dummies.com online articles:** Companion articles to this book's content can be found online at [www.dummies.com/extras/iosappdevelopment](http://www.dummies.com/extras/iosappdevelopment). The topics range from tips on building an interface, using frameworks in iOS app development, and ten ways to make your app-developing life easier, among others.
- ✓ **Updates:** If this book has any Updates after printing, they will be posted to [www.dummies.com/extras/iosappdevelopment](http://www.dummies.com/extras/iosappdevelopment).

## *Where to Go from Here*

If you're starting from the beginning, I would suggest you start with Chapter [1](#). However, if you are brushing up your skills, feel free to jump into a chapter that is particularly relevant to your question. Chapters such as Chapter [10](#), “Adding Animation and Sound to Your App” may be useful to you on their own. The chapters in Part [V](#) are relatively independent of one another.

# Part I **Getting Started**



Visit [www.dummies.com](http://www.dummies.com) for more great content online.