

How To ^{Make an} iPhone App

and start your software business today!



- 7 Lessons to Help You Understand iPhone Development
- Exclusive 12 Step iPhone App Development Formula
- Become an Expert Programmer/Software Business Owner

BONUSES!

- The Objective-C Crash Course
- Ready to Use Source Code for the Done! iPhone App

by Matt Campbell

by Matt Campbell

Legal Notices

This publication is protected under the US Copyright Act of 1976 and all other applicable international, federal, state and local laws, and all rights are reserved, including resale rights: you are not allowed to give or sell this book to anyone else. If you received this publication from anyone other than How to Make iPhone Apps or App Shop, LLC, you've received a pirated copy. Please contact us via e-mail at support@app-shop.com and notify us of the situation.

Please note that much of this publication is based on personal experience and anecdotal evidence. Although the author and publisher have made every reasonable attempt to achieve complete accuracy of the content in this book, they assume no responsibility for errors or omissions. Also, you should use this information as you see fit, and at your own risk. Your particular situation may not be exactly suited to the examples illustrated here; in fact, it's likely that they won't be exactly the same, and you should adjust your use of the information and recommendations accordingly.

Any trademarks service marks, product names or named features are assumed to be the property of their respective owners, and are used only for reference. There is no implied endorsement if we use one of these terms.

iPhone™, iPod™ and iTunes™ are trademarks of Apple Inc., registered in the U.S. and other countries.

Finally, use your head. Nothing in this book is intended to replace common sense, legal, medical, metaphysical or other professional advice, and is meant to inform and entertain the reader. So have fun with *How to Make an iPhone App* and get your stuff done.

Copyright © 2009 App Shop, LLC. All rights reserved worldwide.

Part One: Your First iPhone Application!

Before we dive into the main programming course let's make sure that we have XCode setup correctly. To do that we are going to create a simple application that displays an alert box.

Hello World!

Every programming book seems to start off with the proverbial "Hello World" program. Once you have a Hello World application finished you will be able to setup a new application yourself, write code and test it.

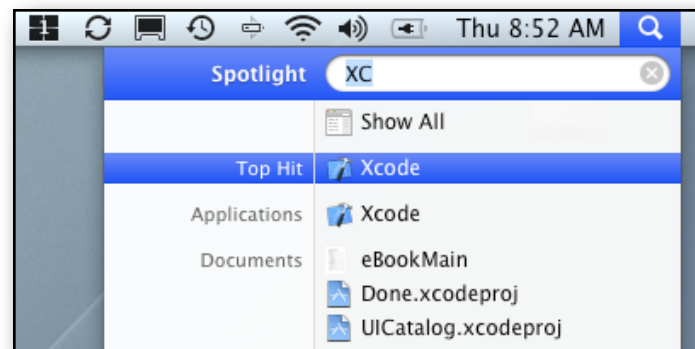
Don't Forget!

Remember to follow the instructions in the introduction and make sure you have a Mac with a working installation of XCode.

Start XCode

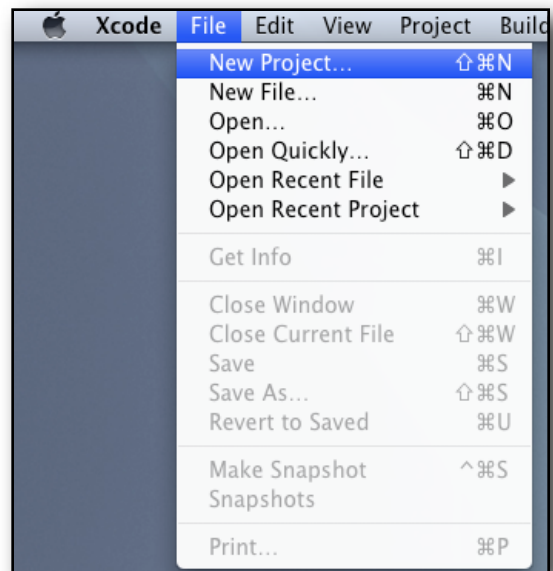
Start up the XCode program on your Mac. If you have not yet signed up for a developer account with Apple and downloaded XCode, see the introduction for instructions on how to do that now.

If you are having trouble finding XCode you can click on the magnifying glass at the upper right hand corner of the your screen to use Spotlight to locate XCode for you.

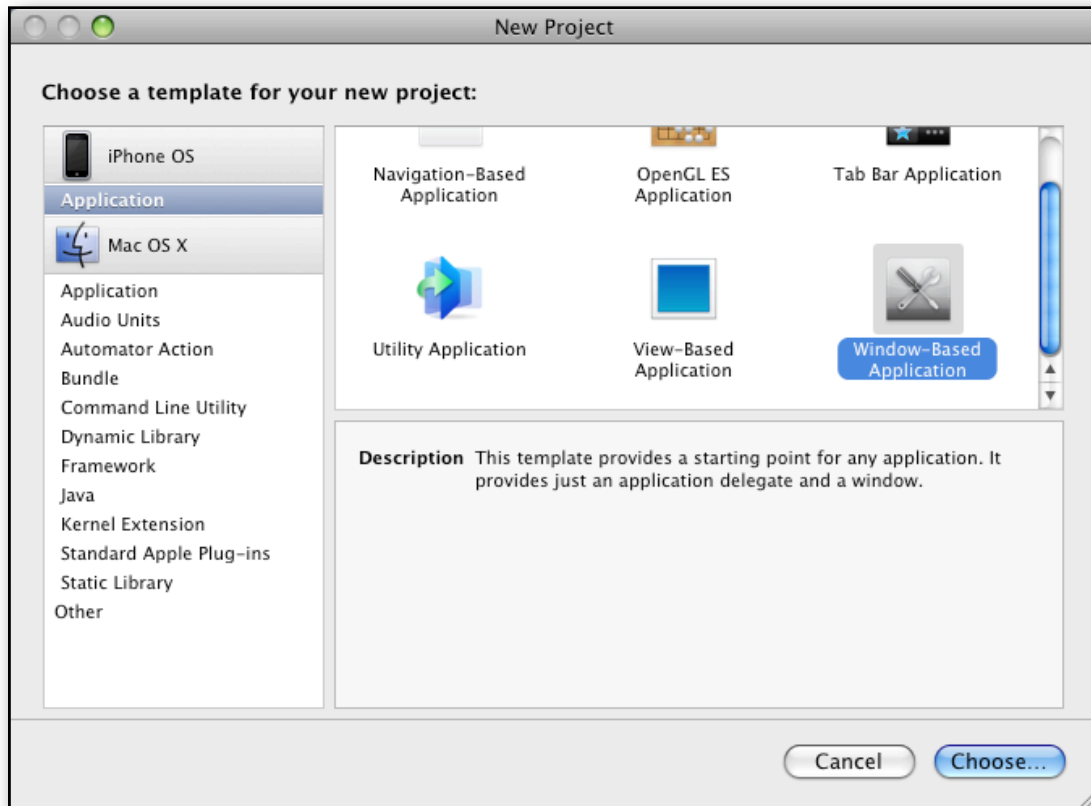


Create a Window-Based Application

Select "File" and then "New Project" from the menu bar at the top of your screen. A dialog box will appear showing you all the options you have for creating iPhone applications.

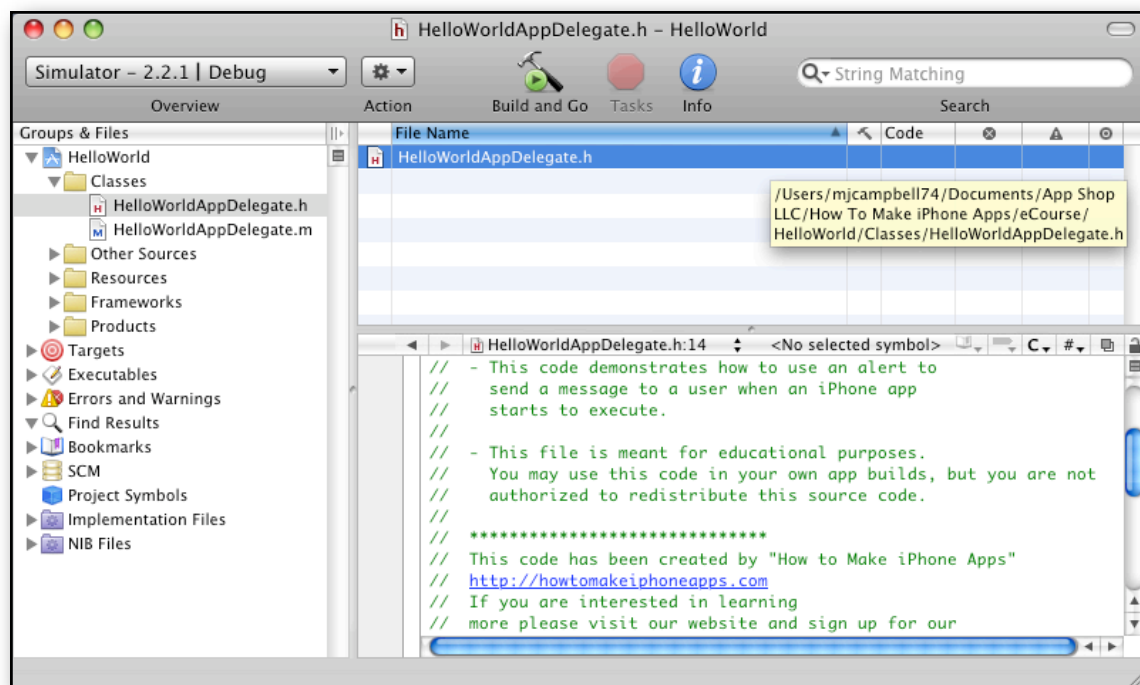


Select "iPhone OS" and then "Window-Based Application". Click the "Choose" button.



A Save As dialog box will appear. Type in the name HelloWorld (no spaces) and press return.

Congratulations! You have created your first iPhone project. The screen that appears is the main XCode editor screen.



This screen is where all your code and other resources are organized for your iPhone applications.

Let's Build It!

Now, take a minute to build this project to make sure that everything is working by pressing the green play button that says "Build and Go" at the top of the XCode window. When you click this button you are instructing XCode to build your application.

Building means turning the code you type into code a computer can use to make applications. XCode automatically loads the application into the iPhone simulator and runs it for you after the application is built successfully.

You should see an image of an iPhone pop up displaying a white screen. This is your Hello World iPhone application that is running in the iPhone simulator.

Now, click on the red octagon labeled "Tasks" at the top of the screen to stop the iPhone application from executing. As you can see, it is easy to create an iPhone project and build the iPhone application. The next step is to make

your new iPhone application your own by having it use your custom content. Here is how you start to do that.

Introducing the Application Delegate

For our purposes, the application delegate is the first place where the code starts to execute. XCode creates the code for the application delegate for you and places the application delegate files under the "Classes" group in your project. You will recognize the files because they will start with your application's name and will have the words AppDelegate attached to the end.

XCode creates an interface file (ends in ".h") and an implementation file (ends in ".m"). Interface files contain the definitions of a class while implementation files contain the code that makes the class work.

The Code Starts to Execute Here

Open up the application delegate implementation file, HelloWorldAppDelegate.m, to see the contents. For now, focus on applicationDidFinishLaunching and the dealloc methods:

```
#import "HelloWorldAppDelegate.h"

@implementation HelloWorldAppDelegate

@synthesize window;

- (void)applicationDidFinishLaunching:(UIApplication *)application {

    [window makeKeyAndVisible];
}

- (void)dealloc {
    [window release];
    [super dealloc];
}

@end
```

applicationDidFinishLaunching is the method that is called before anything else of interest to us right now. dealloc is a method that is used to release objects from memory before the application stops executing. These two

methods are the beginning and the end of your iPhone application's life-cycle.

Hello World Alert!

To get your application to shout "Hello World!" you will use what is called an alert. Here is the pattern: create an alert object from the UIAlertView class, use the alert object's show method and then release the alert object.

Put this code into the applicationDidFinishLaunching method so that the alert box will show itself before anything else happens.

```
#import "HelloWorldAppDelegate.h"

@implementation HelloWorldAppDelegate

@synthesize window;

- (void)applicationDidFinishLaunching:(UIApplication *)application {

    UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"Alert!"
                                                         message:@"Hello World!"
                                                         delegate:self
                                                         cancelButtonTitle:@"OK"
                                                         otherButtonTitles:nil];

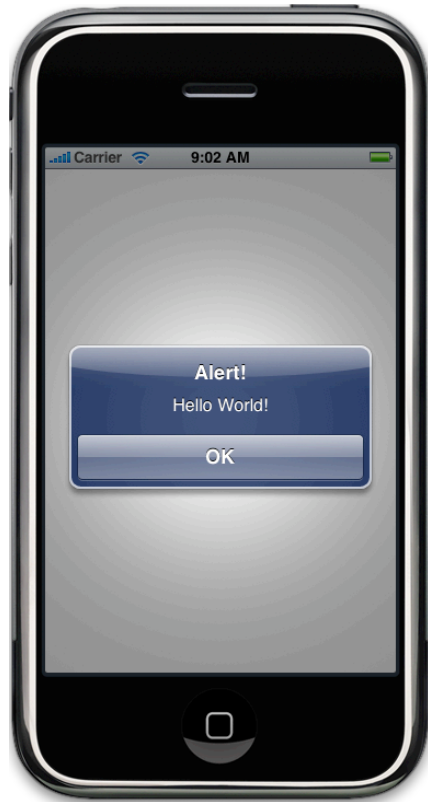
    [alert show];
    [alert release];

    [window makeKeyAndVisible];
}

- (void)dealloc {
    [window release];
    [super dealloc];
}

@end
```

When you Build and Run this code an alert box will be displayed that says "Hello World!"



Stop for a second and take a look at the code that presented here. For our purposes, the `applicationDidFinishLaunching` method is the entry point of the application. The `UIAlertView` object presented here is an example of a way to communicate with the users.

Take a look at how the `UIAlertView` works – it follows a very common pattern in iPhone programming:

- Allocate memory and instantiate an object from a class.
- Use the object.
- Release the object's memory allocation.

Summary

You have now created your first iPhone application!

This first lesson introduces you to some tantalizing details about what makes the iPhone work. However, to really understand how to use these things you will need to learn how to write computer programs.

Once we have a better grounding in programming we will come back to the iPhone and create a more complex application. For now though, it is important to learn how to construct computer programs and organize your code.