

IOS - FIRST IPHONE APPLICATION

http://www.tutorialspoint.com/ios/ios_first_iphone_application.htm

Copyright © tutorialspoint.com

Creating the First App

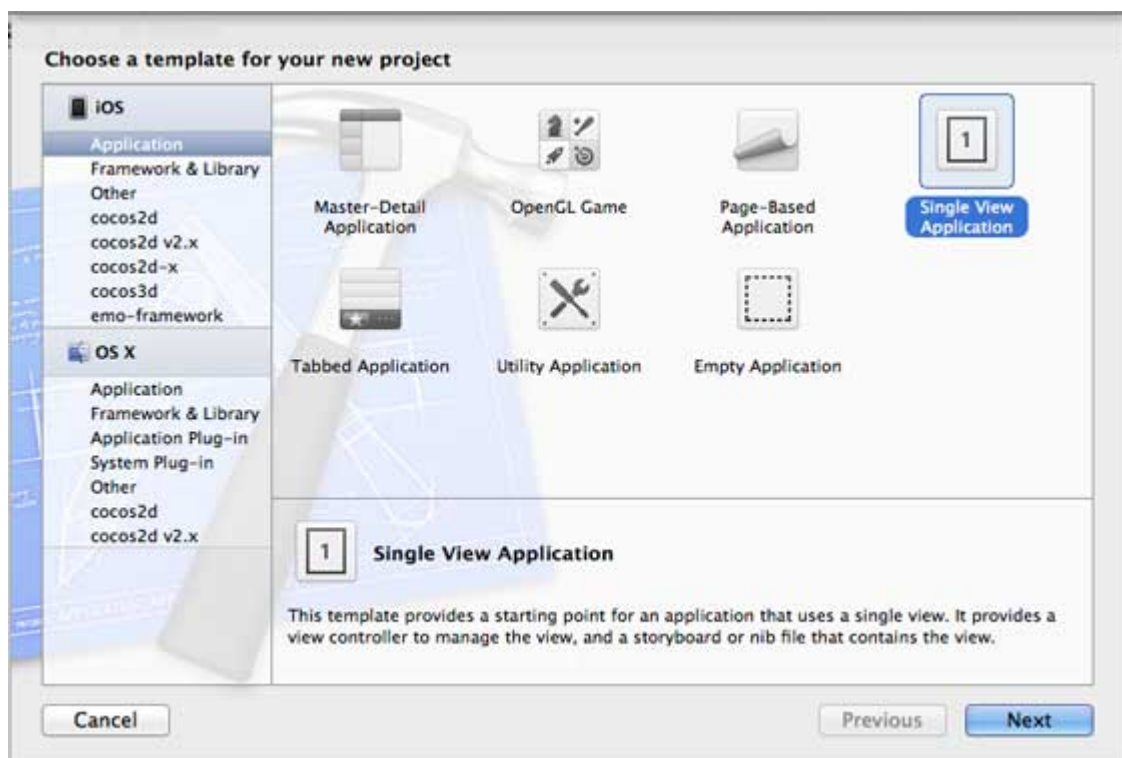
Now we are going to create a simple single view application *ablanapp* that will run on the iOS simulator.

The steps are as follows.

Step 1 – Open Xcode and select **Create a new Xcode project**.

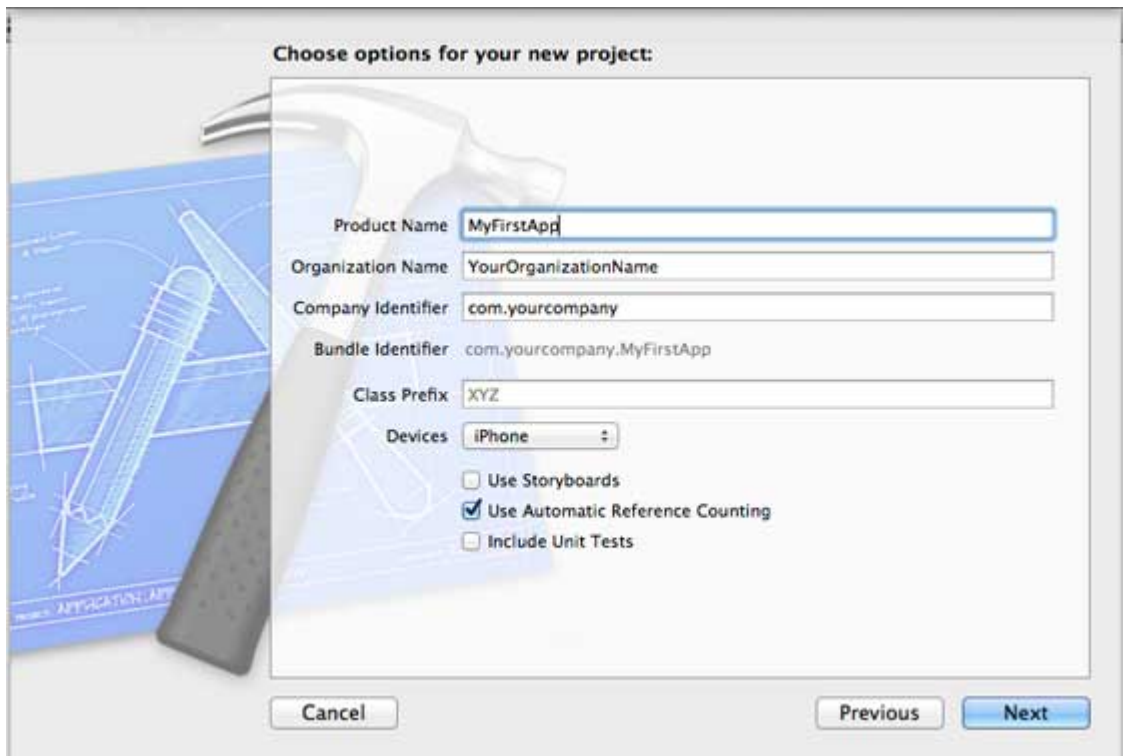


Step 2 – Select **Single View Application**.



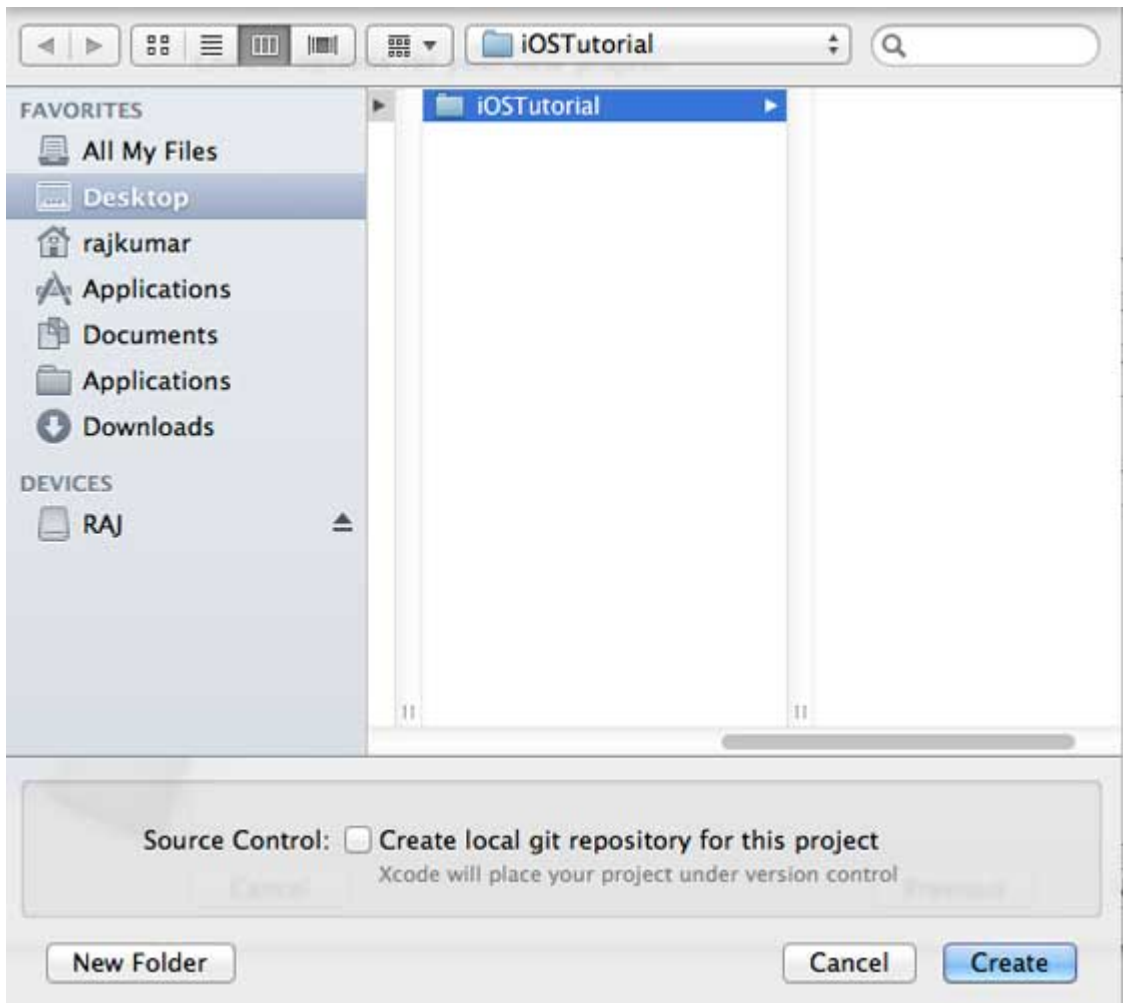
Step 3 – Enter the product name, i.e., the name of the application, organization name, and then

the company identifier.



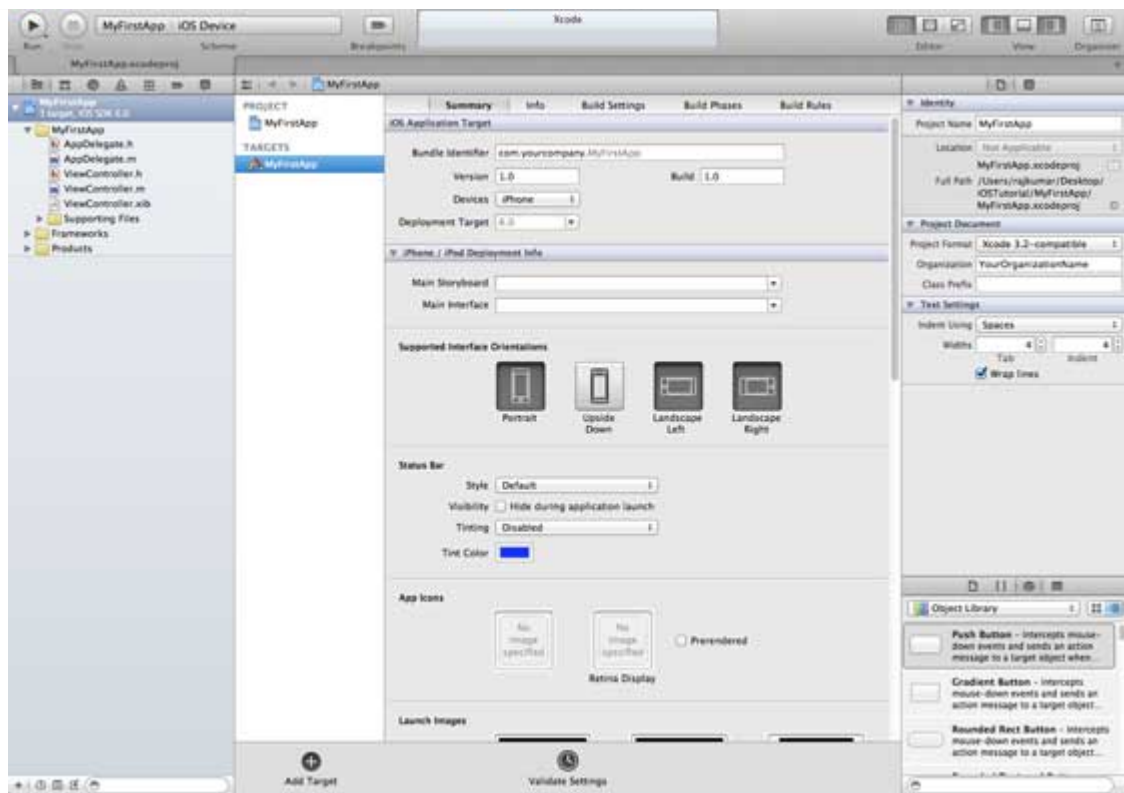
Step 4 – Ensure that **Use Automatic Reference Counting** is selected in order to automatically release the resources allocated once it goes out of scope. Click Next.

Step 5 – Select the directory for the project and select create.



Step 6 – You will see a screen as follows –



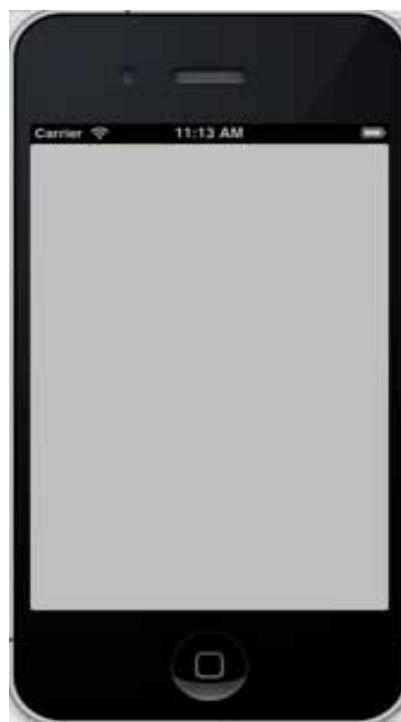


In the screen above, you will be able to select the supported orientations, build and release settings. There is a field deployment target, the device version from which we want to support, let's select 4.3, which is the minimum deployment target allowed now. For now, these are not required and let's focus on running the application.

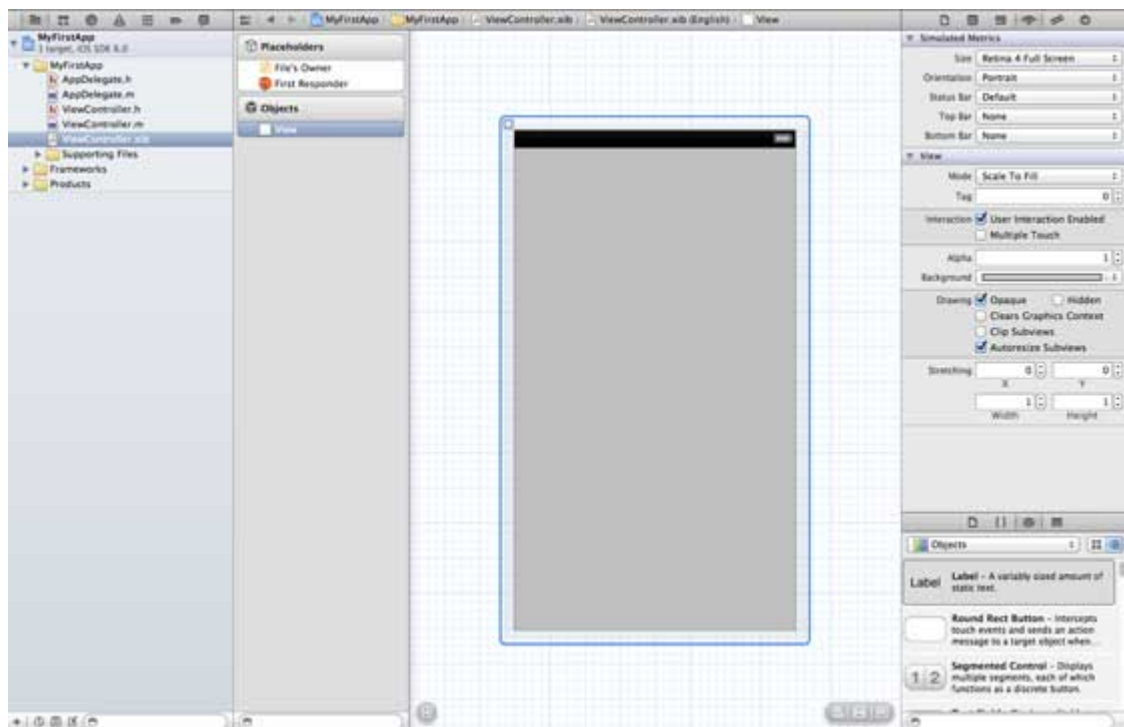
Step 7 – Now, select iPhone simulator in the drop down near Run button and select run.



Step 8 – That's it; you have successfully run your first application. You will get an output as follows –

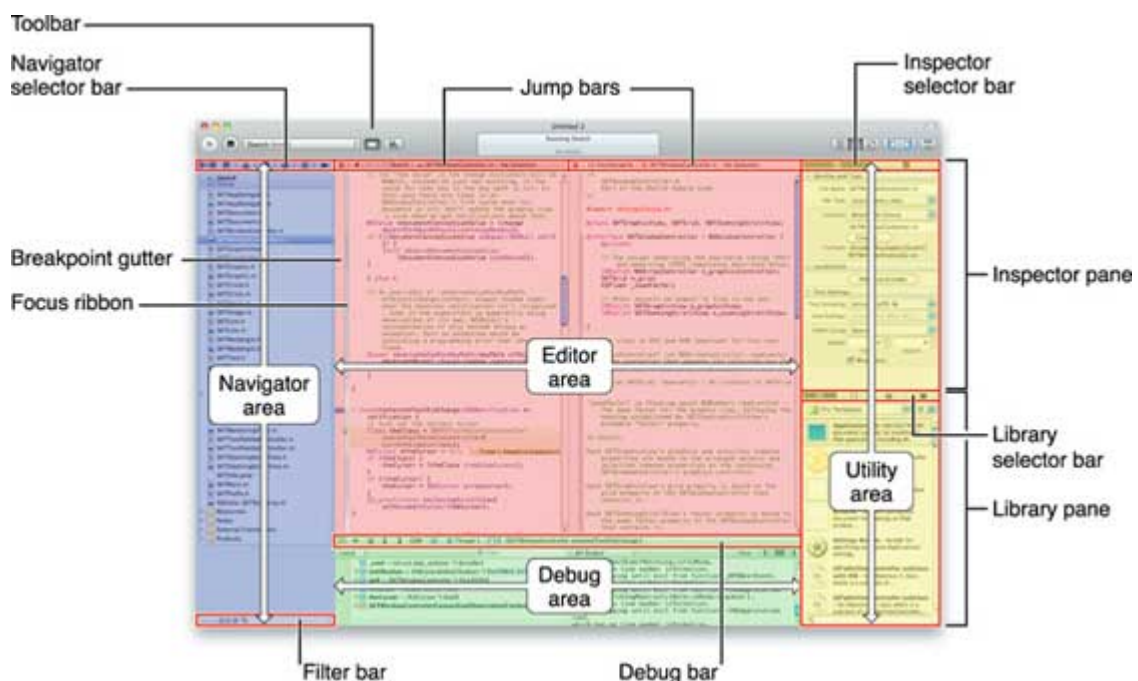


Now let's change the background color, just to have a start with the interface builder. Select ViewController.xib. Select background option in the right side, change the color and run.



In the above project, by default, the deployment target would have been set to iOS 6.0 and auto-layout will be enabled. To ensure that our application runs on devices that are on iOS 4.3 onwards, we have already modified the deployment target at the start of creation of this application, but we didn't disable auto-layout.

To disable auto-layout, we need to deselect the auto-layout checkbox in the file inspector of each nib, i.e., the xib files. The various sections of Xcode project IDE are given in the following figure
Courtesy: AppleXcode4Userdocumentation.



File inspector is found in the inspector selector bar as shown above and auto layout can be unchecked there. Auto layout can be used when you want to target only iOS 6 devices. Also, you'll be able to use many new features like passbook if you raise the deployment target to iOS 6. For now, let's stick to iOS 4.3 as the deployment target.

Code of the First iOS Application

You will find five different files that would have been generated for your application. They are

listed as follows –

- AppDelegate.h
- AppDelegate.m
- ViewController.h
- ViewController.m
- ViewController.xib

AppDelegate.h

```
// Header File that provides all UI related items.
#import <UIKit/UIKit.h>

// Forward declaration (Used when class will be defined /imported in future)
@class ViewController;

// Interface for AppDelegate
@interface AppDelegate : UIResponder <UIApplicationDelegate>

// Property window
@property (strong, nonatomic) UIWindow *window;

// Property ViewController

@property (strong, nonatomic) ViewController *viewController;
//this marks end of interface
@end
```

Important items in code –

- AppDelegate inherits from UIResponder that handles iOS events.
- Implements the delegate methods of UIApplicationDelegate, which provides key application events like finished launching, about to terminate and so on.
- UIWindow object to manage and co-ordinate the various views on the iOS device screen. It's like the base view over which all other views are loaded. Generally there is only one window for an application.
- UIViewController to handle the screen flow.

AppDelegate.m

```
// Imports the class AppDelegate's interface
#import "AppDelegate.h"

// Imports the viewController to be loaded
#import "ViewController.h"

// Class definition starts here
@implementation AppDelegate

// Method to intimate us that the application launched successfully
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];

    // Override point for customization after application launch.
    self.viewController = [[ViewController alloc]
initWithNibName:@"ViewController" bundle:nil];
    self.window.rootViewController = self.viewController;
    [self.window makeKeyAndVisible];
    return YES;
}
```

```

}

- (void)applicationWillResignActive:(UIApplication *)application
{
    /* Use this method to release shared resources, save user data,
    invalidate timers, and store enough application state information
    to restore your application to its current state in case it is
    terminated later. If your application supports background
    execution, this method is called instead of
    applicationWillTerminate: when the user quits.*/
}

- (void)applicationWillEnterForeground:(UIApplication *)application
{
    /* Called as part of the transition from the background to the
    inactive state. Here you can undo many of the changes made on
    entering the background.*/
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
    /* Restart any tasks that were paused (or not yet started) while
    the application was inactive. If the application was previously in
    the background, optionally refresh the user interface.*/
}

- (void)applicationWillTerminate:(UIApplication *)application
{
    /* Called when the application is about to terminate. Save data if
    appropriate. See also applicationWillEnterBackground:. */
}

- (void)applicationWillTerminate:(UIApplication *)application
{
    /* Called when the application is about to terminate. Save data if appropriate.
    See also applicationWillEnterBackground:. */
}

@end

```

Important items in code –

- UIApplication delegates are defined here. All the methods defined above are UI application delegates and contains no user defined methods.
- UIWindow object is allocated to hold the application allocated.
- UIViewController is allocated as the window's initial view controller.
- To make the window visible, makeKeyAndVisible method is called.

ViewController.h

```

#import <UIKit/UIKit.h>

// Interface for class ViewController
@interface ViewController : UIViewController

@end

```

Important items in code –

- The ViewController class inherits the UIViewController, which provides the fundamental view management model for the iOS applications.

ViewController.m

```

#import "ViewController.h"

// Category, an extension of ViewController class
@interface ViewController ()

@end

@implementation ViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

@end

```

Important items in code –

- Two methods implemented here are defined in the base class UIViewController.
- Do initial setup in viewDidLoad which is called after the view loads.
- didReceiveMemoryWarning method is called in case of memory warning.

Loading [MathJax]/jax/output/HTML-CSS/jax.js