

# IOS - ENVIRONMENT SETUP

## iOS - Xcode Installation

**Step 1** – Download the latest version of Xcode from <https://developer.apple.com/downloads/>



The screenshot shows the 'Downloads for Apple Developers' page on the Apple Developer website. The 'Categories' sidebar is expanded, showing 'Applications (11)', 'Developer Tools (152)', 'iOS (11)', 'OS X (52)', and 'OS X Server (9)'. The main list shows various developer tools, with 'Xcode 4.5.2' highlighted. The description for Xcode 4.5.2 states: 'This is the release version of the complete Xcode developer toolkit for Mac, iPhone, iPod touch, and iPad. It includes the iOS 6 SDK and OS X 10.8 SDK. Xcode 4.5.2 requires OS X Mountain Lion or OS X Lion.' Below the description are download links for 'Xcode 4.5.2 Release Notes' (197.22 KB) and 'Xcode 4.5.2.dmg' (56.08 GB). Other items listed include 'Java for OS X Developer Preview 11M4001+10M4001' (Jan 11, 2013), 'Xcode 4.4.1' (Dec 5, 2012), 'Xcode 4.4' (Dec 5, 2012), 'Xcode 4.6 Developer Preview 3' (Dec 3, 2012), 'IOUSBFamily Log Release for OS X 10.7.5' (Nov 28, 2012), and 'Hardware IO Tools for Xcode - Late July 2012' (Nov 26, 2012).

**Step 2** – Double click the Xcode dmg file.

**Step 3** – You will find a device mounted and opened.

**Step 4** – There will be two items in the window that's displayed namely, Xcode application and the Application folder's shortcut.

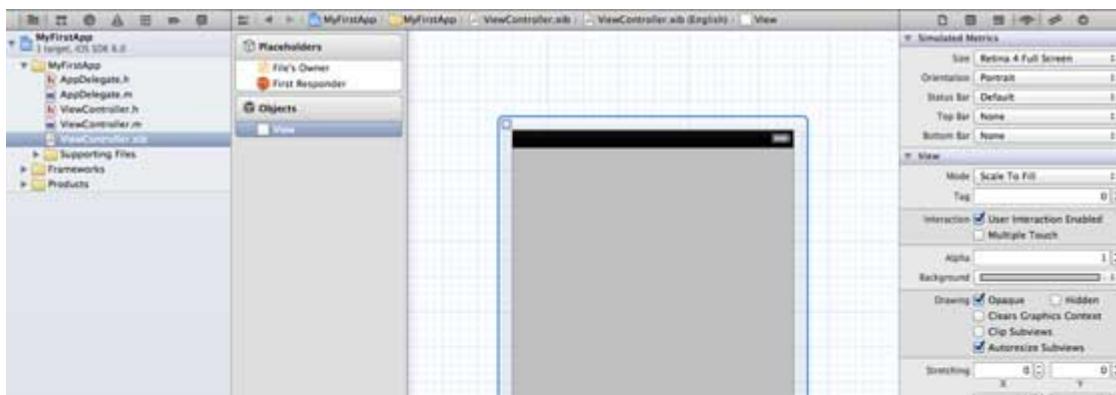
**Step 5** – Drag the Xcode to application and it will be copied to your applications.

**Step 6** – Now Xcode will be available as a part of other applications from which you can select and run.

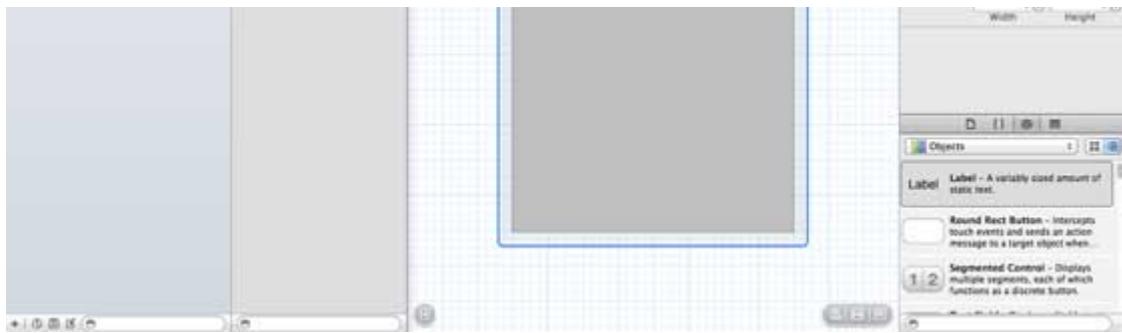
You also have another option of downloading Xcode from the Mac App store and then install following the step-by-step procedure given on the screen.

## Interface Builder

Interface builder is the tool that enables easy creation of UI interface. You have a rich set of UI elements that is developed for use. You just have to drag and drop into your UI view. We'll learn about adding UI elements, creating outlets and actions for the UI elements in the upcoming pages.



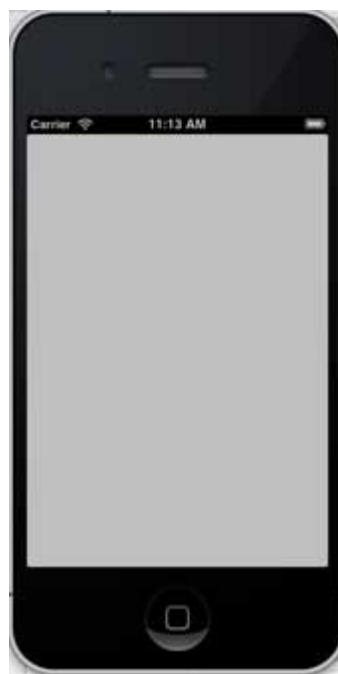
The screenshot shows the Interface Builder interface. On the left is the 'File Navigator' showing a project named 'MyFirstApp' with files like 'AppDelegate.h', 'AppDelegate.m', 'ViewController.h', 'ViewController.m', and 'ViewController.xib'. Below it is the 'Document Outline' showing 'Placeholders' and 'Objects'. The main area shows a 'View' with a black background. To the right are two panels: 'Simulated Metrics' and 'View'. The 'Simulated Metrics' panel shows settings for 'Retina 4 Full Screen', 'Orientation: Portrait', 'Status Bar: Default', 'Top Bar: None', and 'Bottom Bar: None'. The 'View' panel shows settings for 'Mode: Scale To Fill', 'Tag: 0', 'Interaction: User Interaction Enabled (checked)', 'Alpha: 1.0', 'Background: [color bar]', 'Drawing: Opaque (checked)', 'Stretching: X: 0 Y: 0 Z: 1'. There are also checkboxes for 'Hidden', 'Clears Graphics Context', 'Clip Subviews', and 'Autosizes Subviews'.



You have objects library at the right bottom that consists the entire necessary UI element. The user interface is often referred as **xibs**, which is its file extension. Each of the xibs is linked to a corresponding view controller.

## iOS Simulator

An iOS simulator actually consists of two types of devices, namely iPhone and iPad with their different versions. iPhone versions include iPhone *normal*, iPhone Retina, iPhone 5. iPad has iPad and iPad Retina. A screenshot of an iPhone simulator is displayed below.



You can simulate location in an iOS simulator for playing around with latitude and longitude effects of the app. You can also simulate memory warning and in-call status in the simulator. You can use the simulator for most purposes, however you cannot test device features like accelerometer. So, ~~you might always need an iOS device~~ to test all the scenarios of an application thoroughly.

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