

Getting Started with Apple® iOS Development – Apple® iOS SDK – Objective-C

Overview

This document describes the end to end process of designing, packaging, deploying and running an Apple® iPhone/iPod application using the ZebraLink SDK

The sample code used in this guide is from the [MyFirstZebraApp](#) sample code article.

Target Audience

The information delivered in this document assumes the reader has reasonable technical competence covering Apple®iOS environment, core programming concepts and rationales.

System Prerequisites

You must install all the system prerequisites listed in this section.

Note: The default installation options for all these prerequisites are satisfactory.

MAC OS X

Unlike development for other mobile device environments, development for Apple®iOS based mobile devices must be conducted on Mac OS X.

Refer to [Mac OS X](#).

XCode and iOS SDK

XCode is the development environment for apps that run on the Apple® iOS operating system.

For more information and to download links, refer to for [XCode and the iOS SDK](#).

Zebra Multiplatform SDK

The ZebraLink for Apple® iOS digital devices SDK contains all the required components to develop applications for Zebra label printers. The SDK includes the header files to scan for and connect to network based Zebra label printers.

For more information, including system prerequisites and downloads, refer to [ZebraLink SDK](#),

Zebra Network-Enabled Label Printer

In order to fully test the application that you create, you must have a network-enabled Zebra printer capable of understanding ZPL.

For more information, refer to [Zebra label printers](#).

Apple® iOS Device

While the Apple® iOS simulator included in the XCode toolkit satisfies most of the anticipated requirements for developing with Apple® iOS, we recommend that you test all of your development with a physical device.

Creating an Apple® iPhone /iPod application using the Zebralink SDK

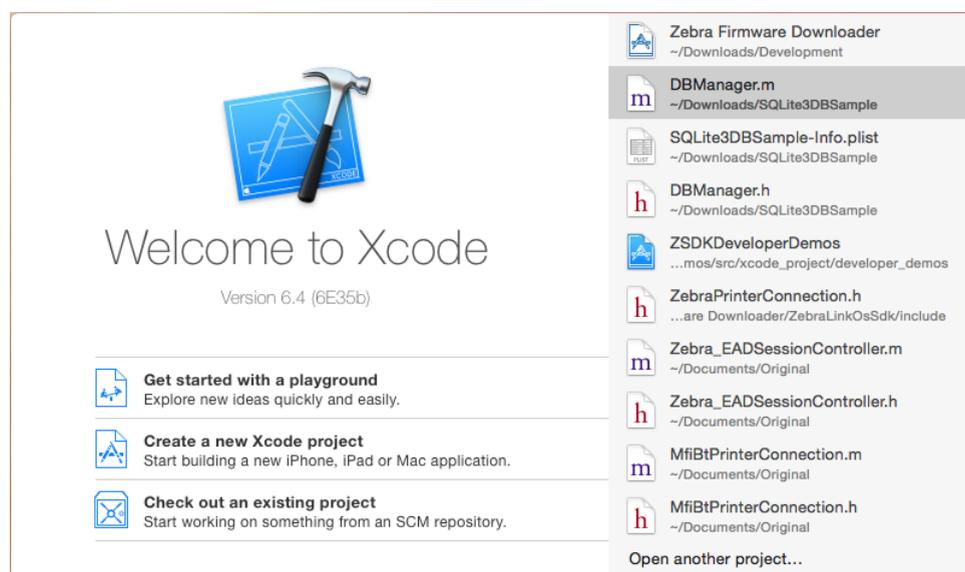
You will be guided on how to create a mobile application in the following sections:

- [Designing a New XCode Project](#)
- [Building the User Interface](#)
- [Linking the User Interface to Your Project](#)
- [Configuring Your Project to Include the ZebraLink SDK](#)
- [Building and Running Your Project](#)

Designing a New XCode Project

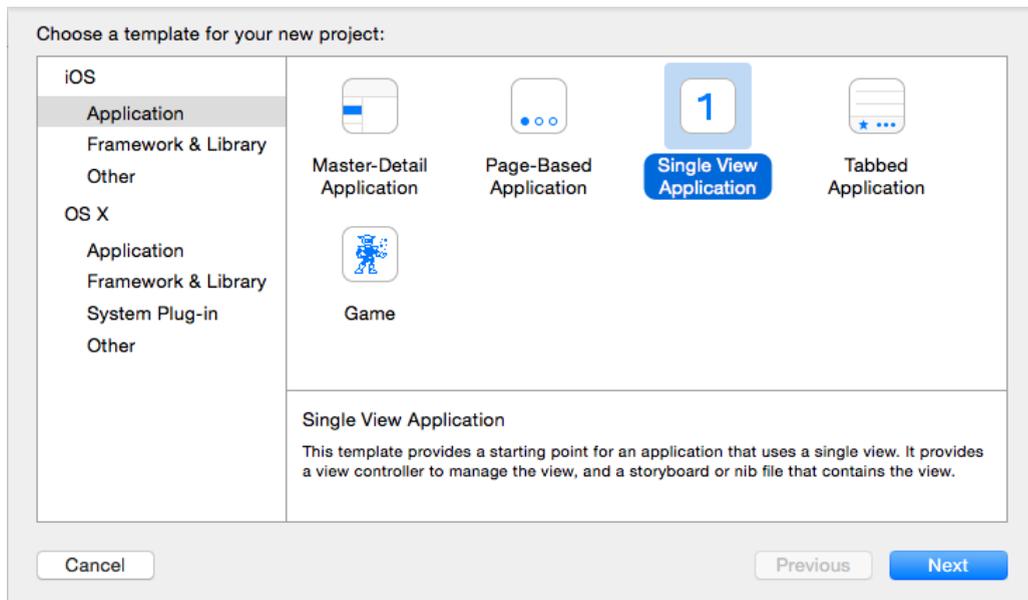
1. Start XCode.

The XCode welcome window appears.



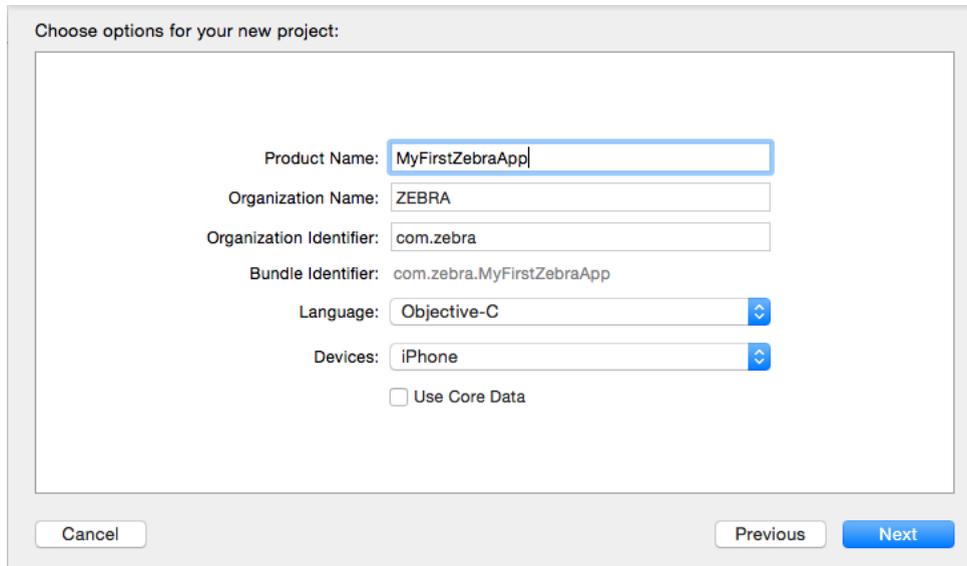
2. Click **Create a new XCode project**.

The template window appears.



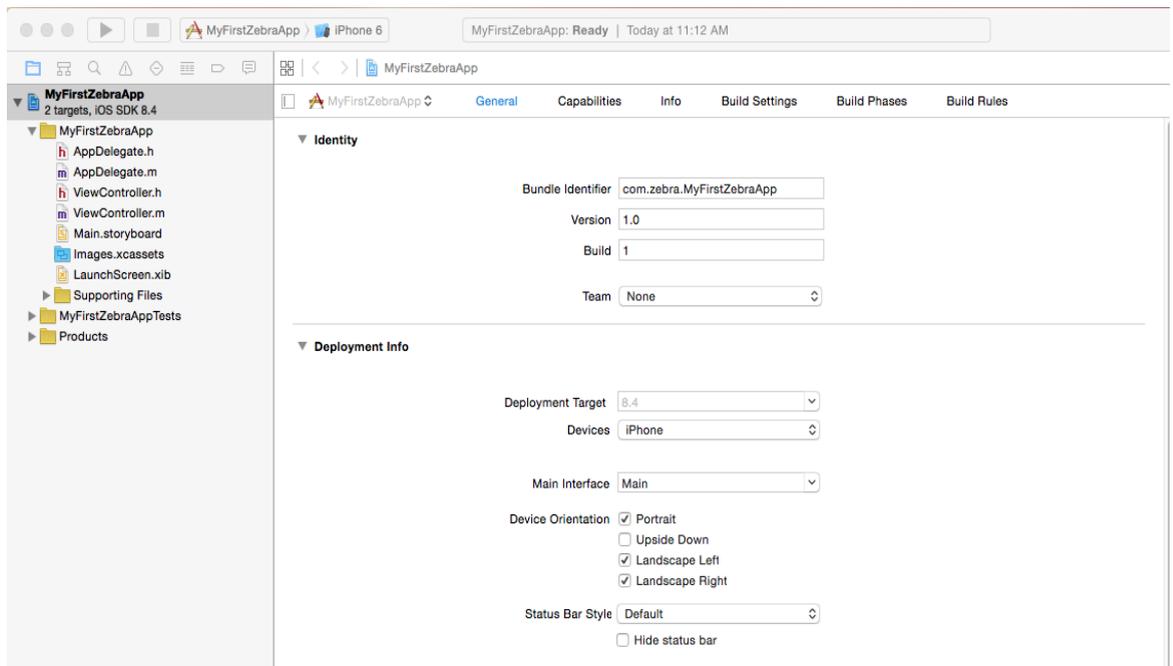
3. Click **Single View Application** from the iOS Application section and click **Next**.

A window appears allowing you to name your new project.



4. Type "MyFirstZebraApp" as your project name and navigate to the location where you want to save your project.
5. Click **Next**.

XCode opens having prebuilt some source files for you.

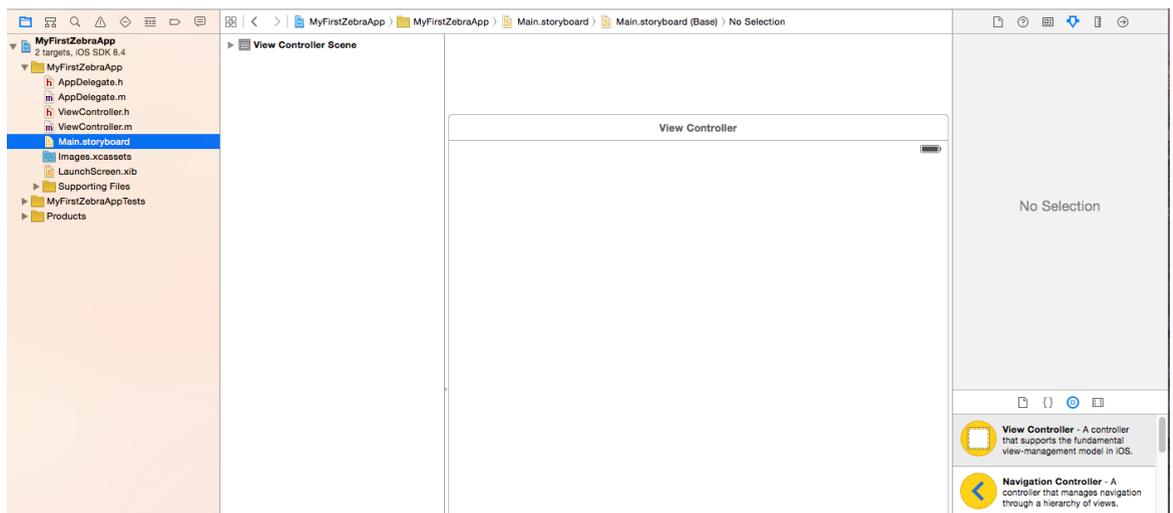


Building the User Interface

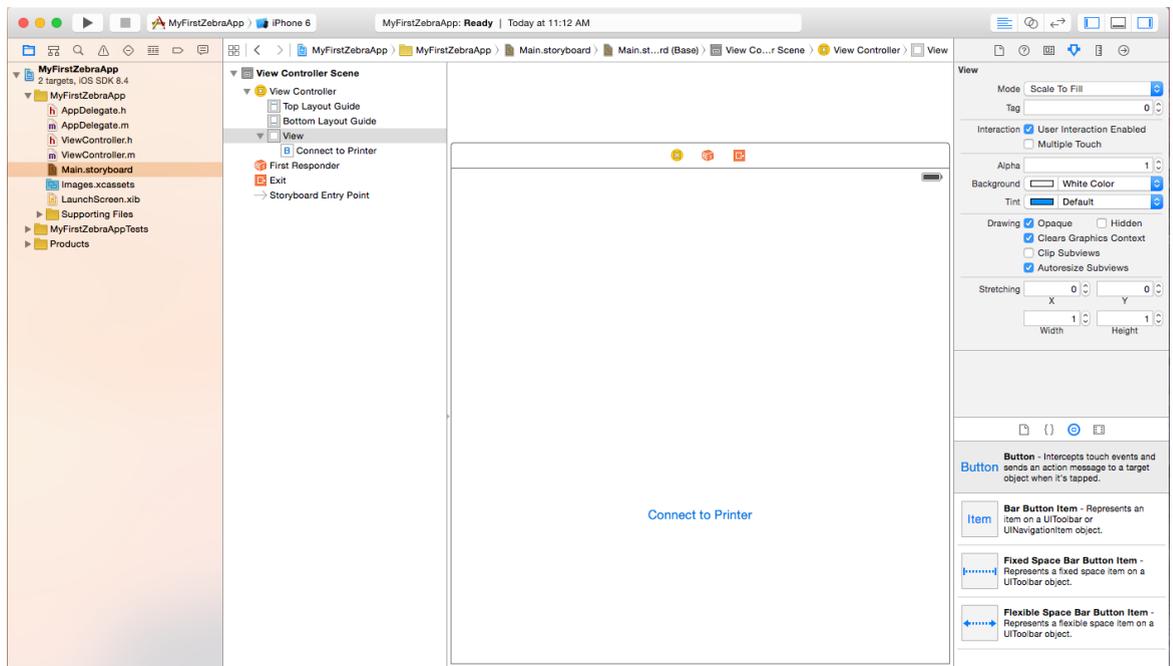
1. Locate and double click the “Main.storyboard” file.

The application opens showing various windows including a:

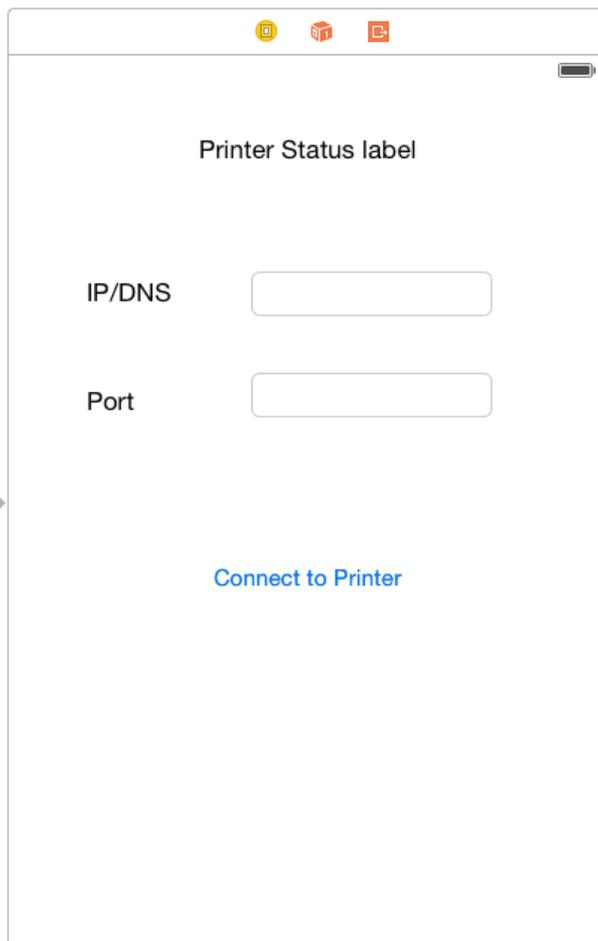
- Library of visual components
- Blank design canvas
- Property window



2. Locate then drag and drop the **Button** item from the Library onto the blank View canvas.
3. Double click the button once it is on the canvas and title it “Connect to Printer”.



4. Add 2 text fields and 3 label components to the canvas as the following dialog displays.



5. Close the Interface Builder and return to XCode.

Linking the User Interface to Your Project

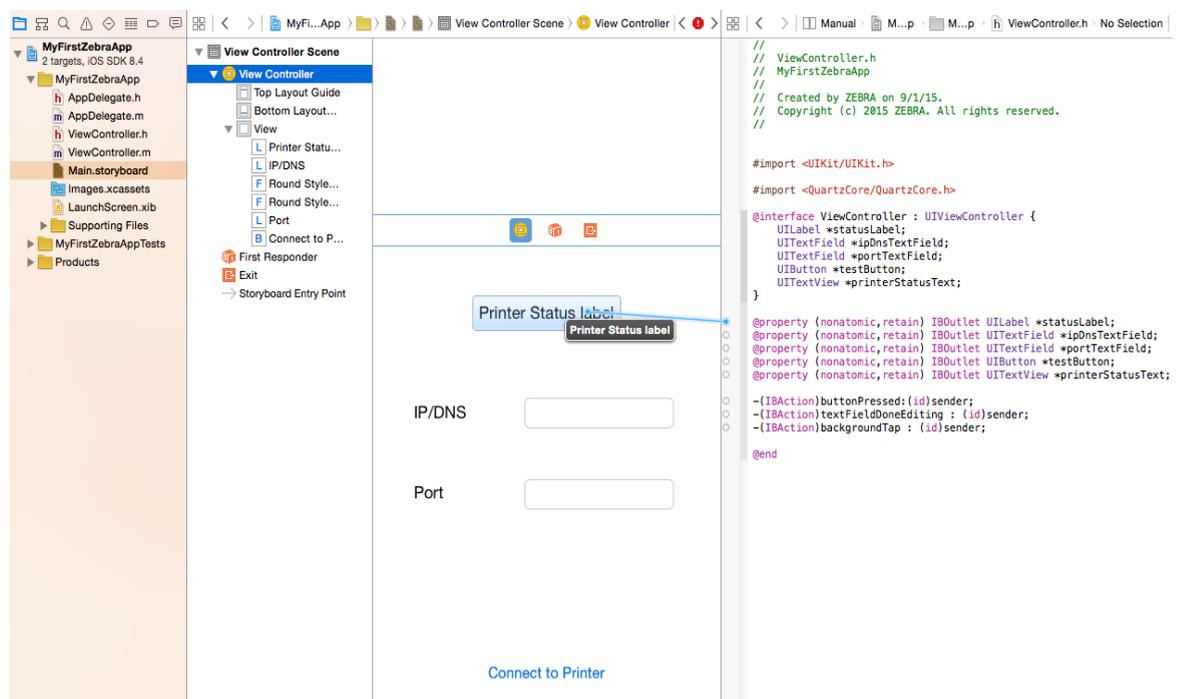
You must add the code to the main header file (ViewController.h) in order to support the components of the user interface.

1. Replace the code in the “ViewController.h” with the contents of the “MyFirstZebraAppViewController.h” file from the “Connection Over IP”. Make sure to update the class name.

Note: The sample code used in this guide is from the [MyFirstZebraApp](#) sample code article.

For information on joining the [ISV Program](#), contact a Program Manager in your area.

2. To now link the components of the user interface to the objects declared in the header file, open the “Main.storyboard” file again to open up the Interface.
3. Drag and drop the property status label from the ViewController.h to the printer status label in the “main. Storyboard”.



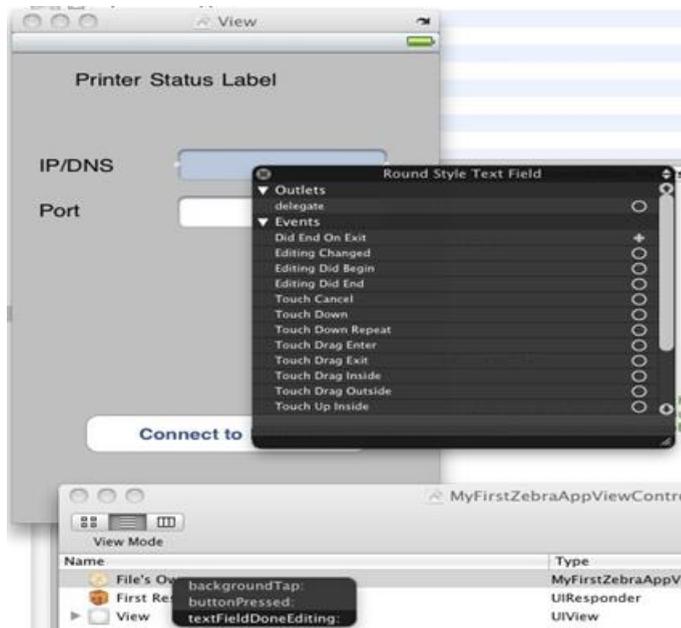
Repeat this process for the remaining properties.

Note: In the similar way, make sure the button pressed event [ViewController.m] is linked to the “Connect to Printer” button on the user interface.

The text fields will be used to enter the IP/DNS name and port for printer. In order that the on-screen keyboard closes when the user finishes entering these settings, the “Did End On Exit” event for each text field must be handled.

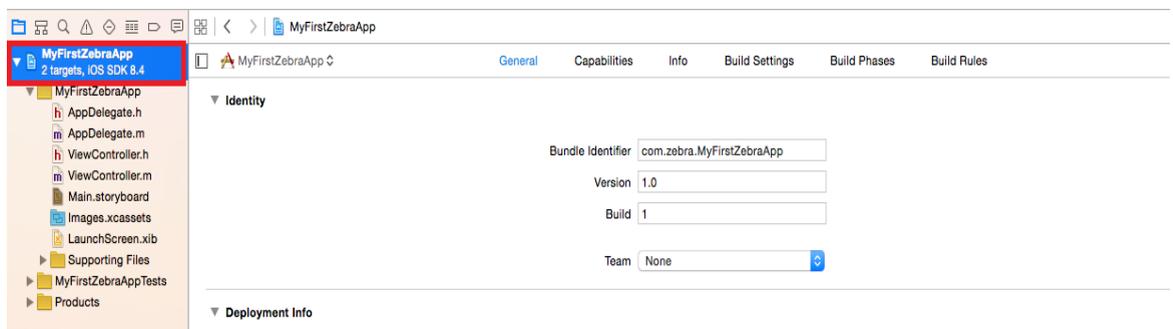
4. Right click on the **IP/DNS** text box.
5. Drag the circle mark next to the “Did End On Exit” event onto “File’s Owner” and choose the “textFieldDoneEditing”.
6. Repeat this process for the Port text field.

7. Close the Interface Builder.

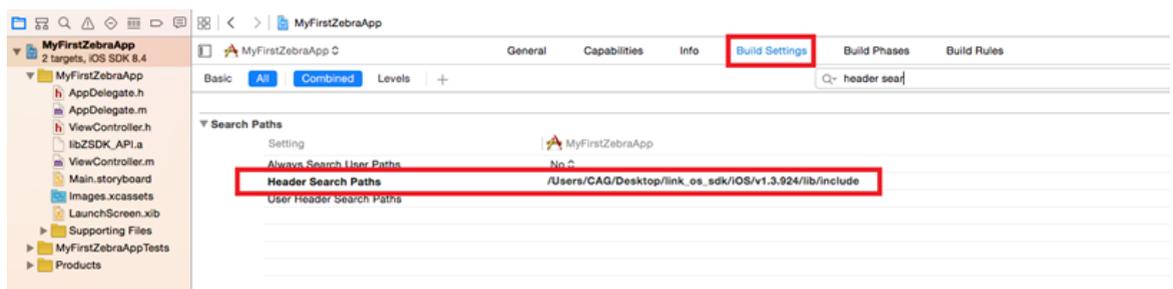


Configuring Your Project to Include the ZebraLink SDK

1. Click the MyFirstZebraApp.
2. Click the Build Settings tab to specify the build information.



3. Search for “Header Search Paths” by typing into the filter box at top of the window.



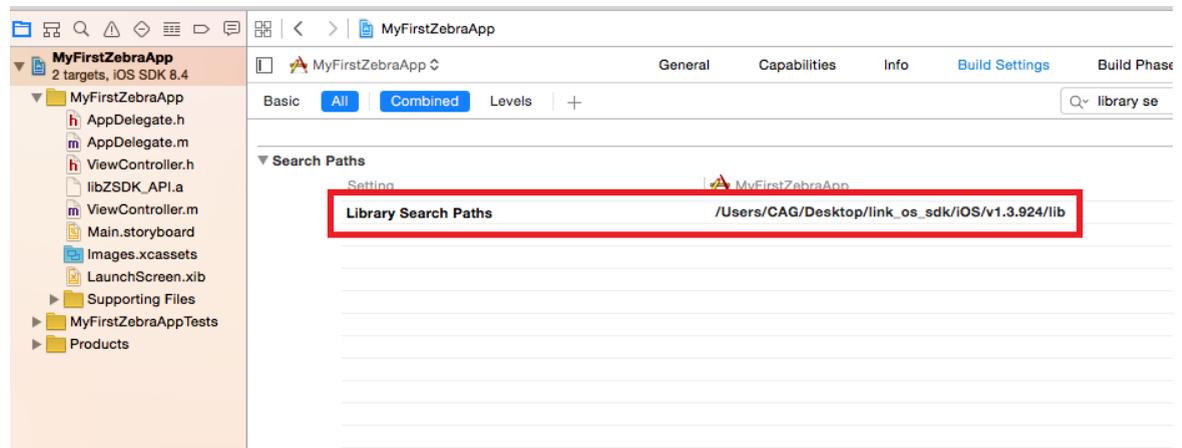
4. Edit the “Path” setting to link to the “include” folder in your installation path.

For instance, if you have installed the ZebraLink SDK to the default folder and your username is “macuser” then your path would be:
 /Users/macuser/Applications/zebralink_sdk/iOS/v1.3.924/lib/include

Note: DO NOT tick the “Recursive” option.

5. Click **OK**.

You must now edit the “Library Search Paths”.

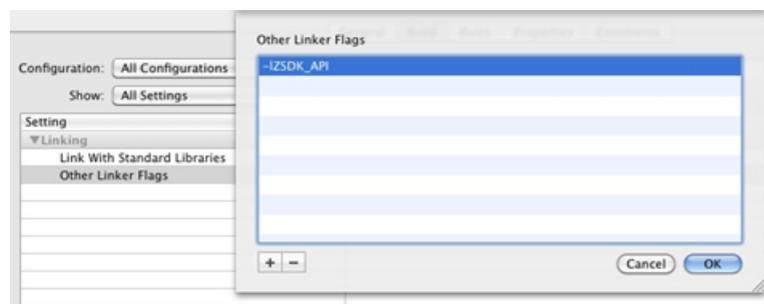


6. Edit the “Path” setting to link to the “lib” folder in your installation path.

For instance, if you have installed the ZebraLink SDK to the default folder and your username is “macuser” then your path would be:
 /Users/macuser/Applications/zebralink_sdk/iOS/v1.3.924/lib

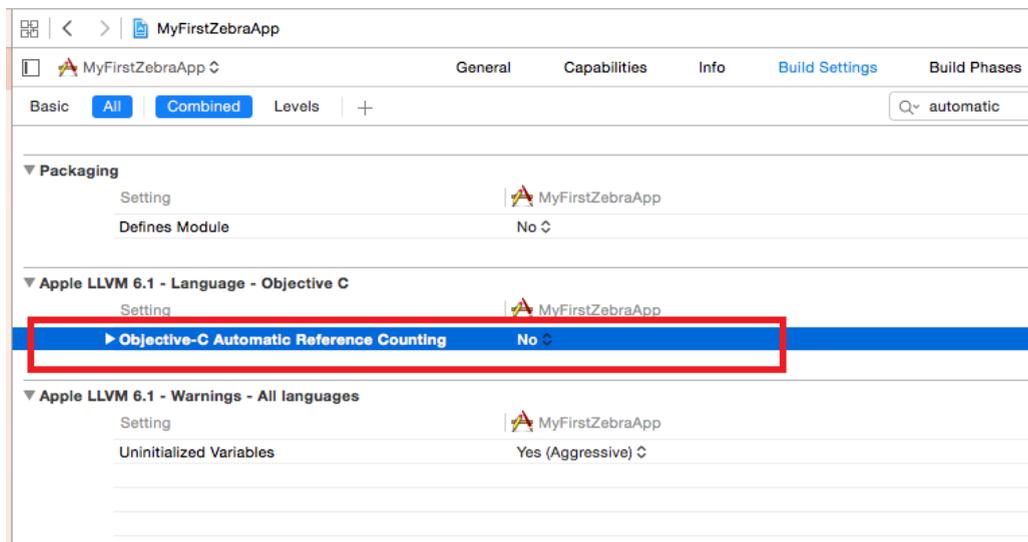
Note: DO NOT tick the “Recursive” option.

7. Click **OK**.
8. Locate the “Other Linker Flags” subheading and add the item “-LZSDK_API”.



Note: You can choose to ignore this step if you directly drag and drop the libZSDK_API.a to the project folder.

You may need to disable the Automatic Reference Counting since you are using the implemented code.



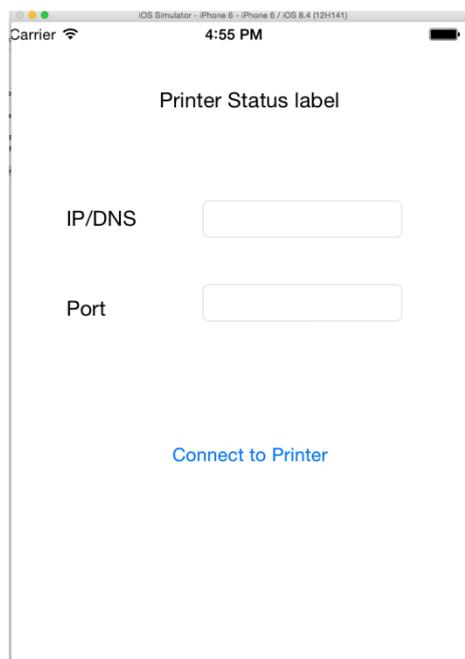
9. Replace the contents of the “ViewController.m” file with the source code from the “MyFirstZebraAppViewController.m” file from the “Connection Over IP”.

Note: Note: The sample code used in this guide is from the [MyFirstZebraApp](#) sample code article.

With all the source code now added, you can build and run the solution.

Building and Running Your Project

Enter the IP/DNS address and Port number for the network printer and click the **Connect to Printer**.



A test label prints.

The “Printer Status Label” updates to reflect the connection and communication status with the printer.

Appendix

See Also

- [ZPL Programming Guide](#)
- [CPCL Programming Guide](#)
- For any further information, sample code and solutions or to request further content, visit the [Zebra Developer Portal](#).

Document Control

Version	Date	Description
1	03-Dec-10	Initial Release
1.1	01-Mar-11	Minor updates to see also section
2.0	28-Jun-12	Updated Chinese link into document
3.0	31-Jul-12	Updated Spanish and Portuguese link into document
4.0	September, 2015	Updated to match with latest iOS XCode
5.0	2nd October 2017	Updated URLs

Disclaimer

All links and information provided within this document are correct at time of writing.

Created for Zebra Global ISV Program by Zebra Development Services.