

Delly Shop Ecommerce Application Template Documentation



<https://www.dropbox.com/s/0dtrb58mit53hho/Screen%20Recording%202019-12-11%20at%2009.29.30.mov?dl=0>

<https://youtu.be/BDVmLuhkq9U>

Created: 12/10/2019

Latest update: 12/10/2019

By: [XamDesign](#)

Support Skype/Email : xamappdesign@gmail.com

DellyShop is an e commerce mobile application design.

The design was written with Xaml code.

The purpose of the application is to save customers dozens of hours spent on the design, so that the time spent on the design will be spent on the backend.

The application does not only consist of XAML code, the Xaml code is filled with a logical architecture on the back.

Simple and beautiful Template structure, CustomRenderers, CustomViews, MVVM architecture design contains.

DellyShop has a simple design and infrastructure.

The application contains very well designed design structure and explanatory codes.

DellyShop supports all language structures regardless of RTL or LTR. It has Arabic and English language structure.

Why Should I Buy This Template?

DellyShop will save you a great deal of time with its superior design and infrastructure features.

Design is a time-consuming and difficult process in Xamarin Forms. Most super developers lose a lot of time with the Design part, but DellyShop will save you time.

DellyShop will never put you in a mold with its flexible and dynamic structure.

Everything is easy, everything is simple! Even if you don't have any knowledge about Xamarin, we are ready to help you with our explanatory documents and our 24/7 support line.

Simply send us your purchase code for support.

If this code is verified by us, we will be happy to support you!

Installation Guide

1. Development Tool Installation

The installation is pretty easy, please follow the steps below:

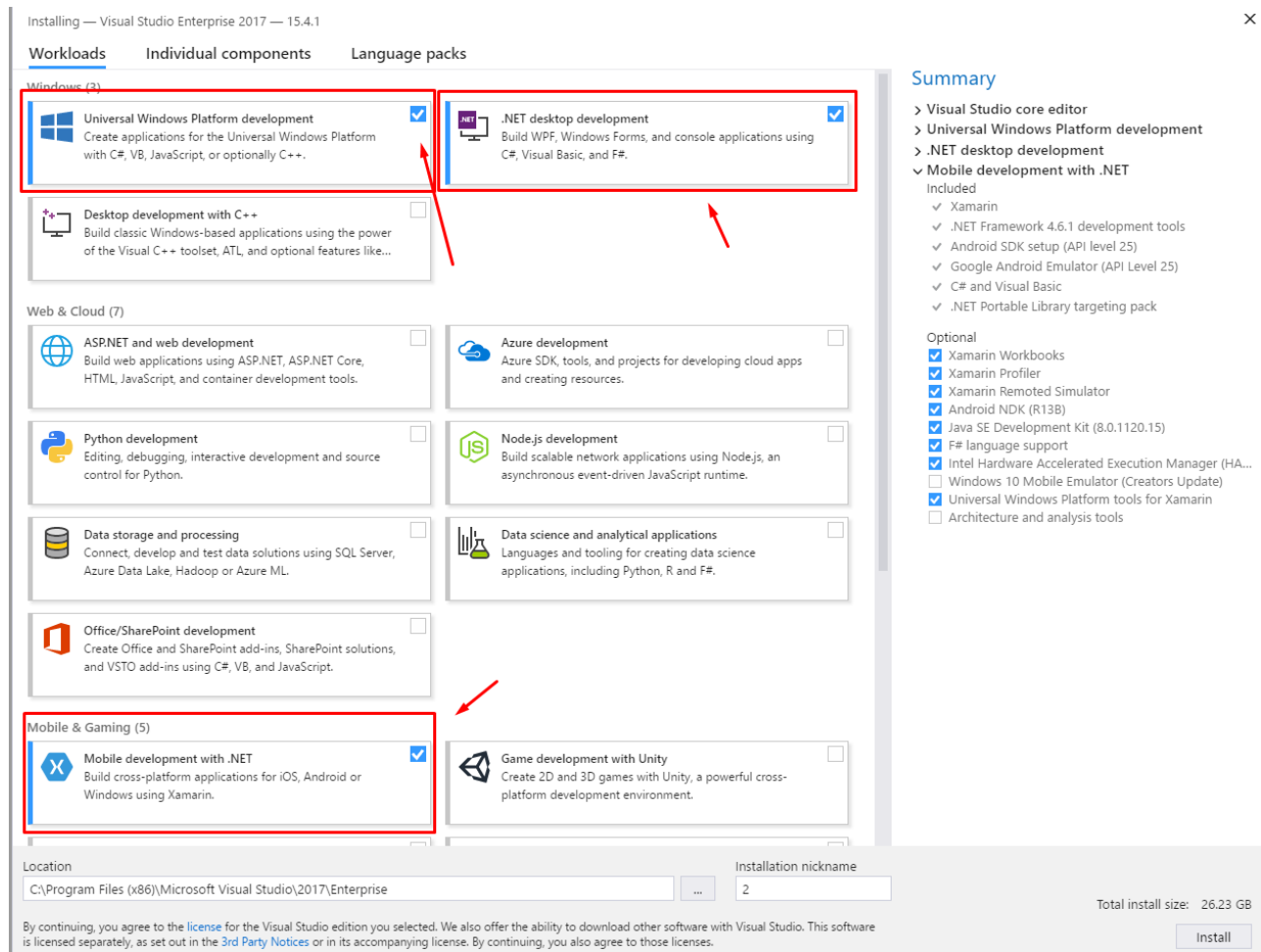
- **DellyShop App** bundle, you can get it from [Here](#)
- Download Visual Studio With Xamarin 2017 [Download Here](#)

- Unzip the **DellyShop** archive, extract it to new folder, and then open the folder.
- In the main folder you will find the solution (Name: **DellyShop** Type: Microsoft Visual Studio Solution) double click on it and wait till everything is loaded.
- In the Visual Studio Build > Clean Solution, click it and wait till it finish.
- Again In the Visual Studio menu bar go to Build > Build Solution, click it and wait till it finish building your project.

a) **Download Visual Studio for Windows as shown below**

https://youtu.be/cOfAl_yv4XM

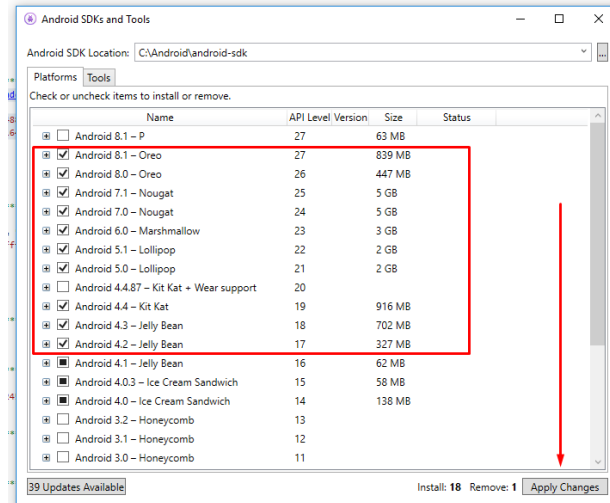
Select This 3 check boxes as the image bellow and then press install for (Visual studio on Windows)



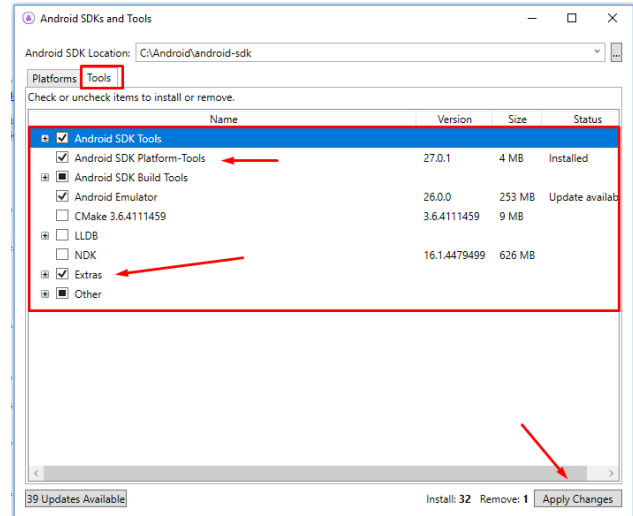
b) Install all Android SDKs in your system.

From your Visual studio go to tools menu >> Android >>Sdk Manager

Select all the Sdk from version 4.3 till 8.1 as images bellow



Sdks Versions which should be selected



Build tools which should be selected

After you installed all the SDKs in your system, you are ready to start and build your app,

Follow step 3.

c) Press on **Build Solution** Then on **Clean Solution**, click on it and wait till it finish.

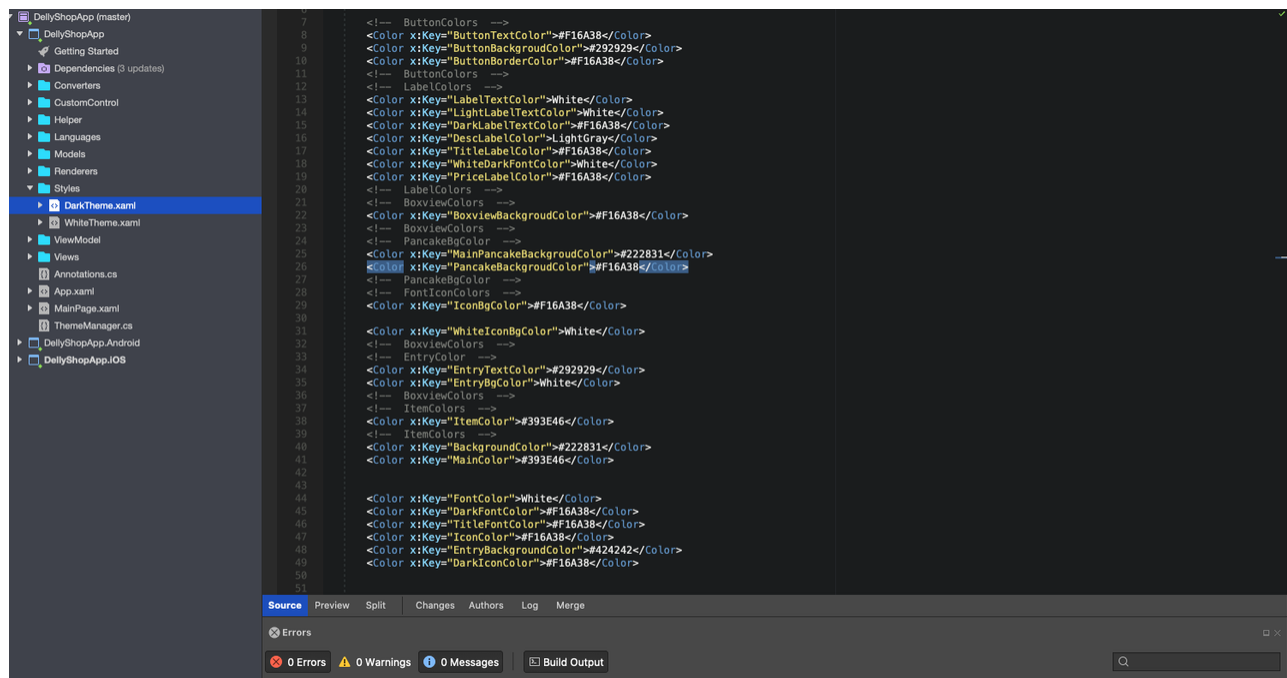
Be sure when your build your application to set your linking to **“SDK and assemblies only”** in either Debug or Release mode before the archive is done or during your test.

Customization Guide

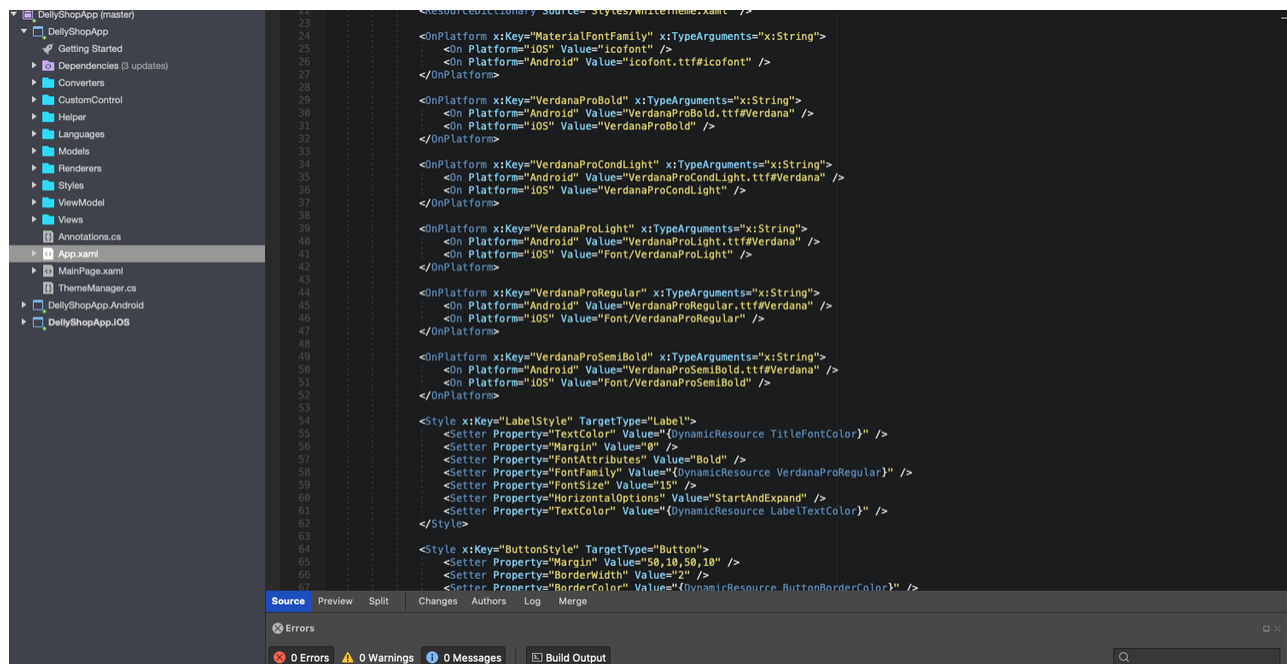
Follow the steps bellow to customize your application as you want

1-Change Colors, Theme and fonts use in the apps

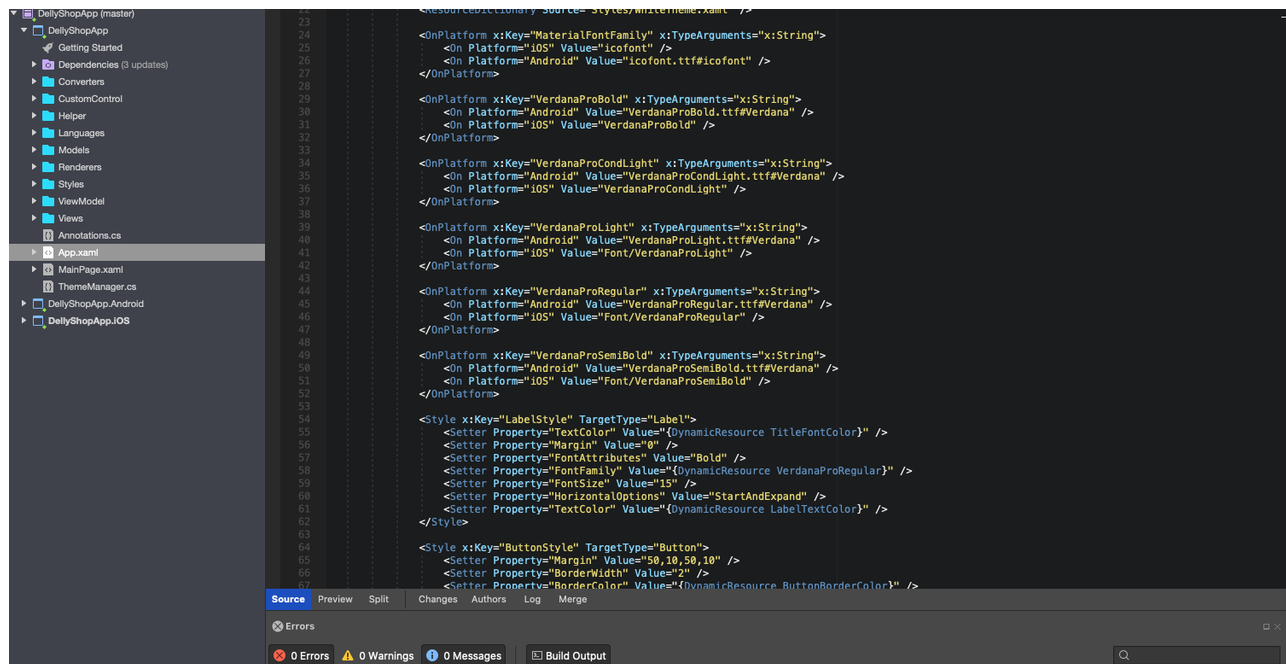
1-Go to **Shared project (DellyShopApp folder) > (Styles) > open DarkTheme.xaml or LightTheme.xaml** file and you will see in the xaml file all customization you need relating to color, theme of the application.



1-Go to **Shared project (DellyShopApp folder)** > open **App.xaml** file and you will see in the xaml file all customization you need relating to Font, Style of the application.



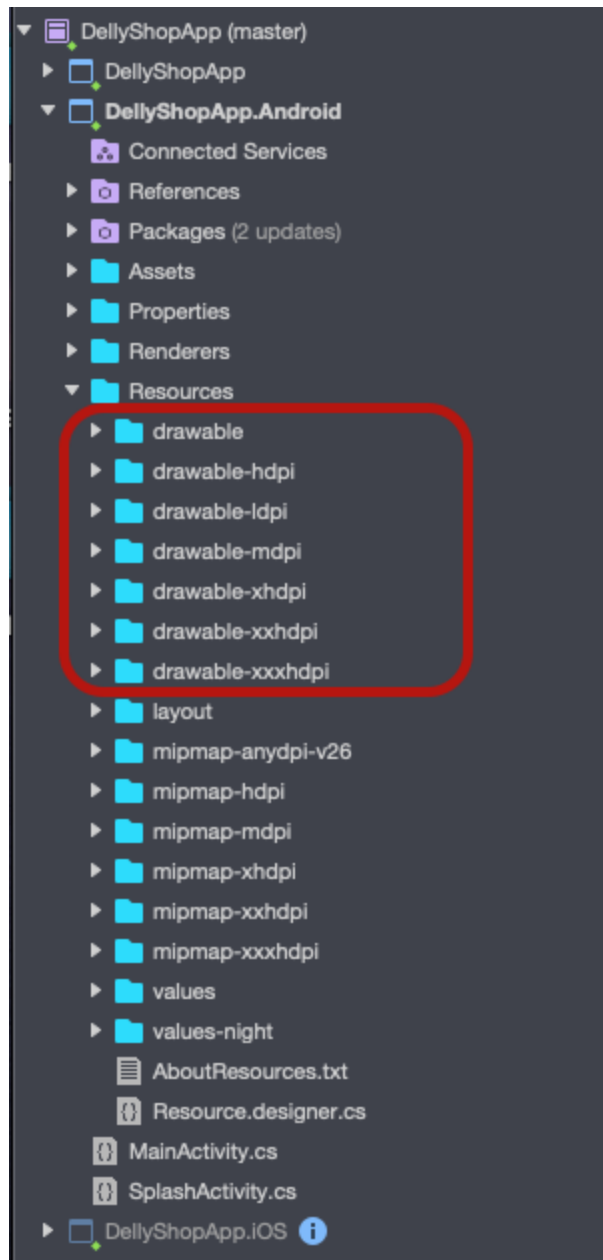
2- For Android application, Go to **Resources > values folder** > open **styles.xml** + **colors.xml** file and you will see all the xml file which is responsible for android app of colors and theme styles ...



2-Change Logos & Icons and images

1 - For Android, in the **DellyShopApp.Android** project you will find 4 main folders by the following names

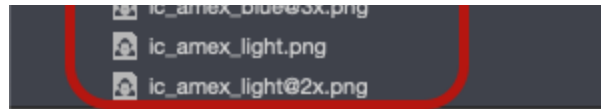
- **drawable** >> for all android screen sizes by default
- **drawable-hdpi** >>for small devices screen
- **drawable-xhdpi** >> for normal devices screen
- **drawable-xxhdpi** >> for High HD screens such as Samsung s8 and s7



2- For iOS, in the **DellyShopApp.iOS** project you will find the **Resources** folder which includes all images & icon assets for iOS application.

- ▼ DellyShopApp (master)
 - ▶ DellyShopApp
 - ▶ DellyShopApp.Android
 - ▼ DellyShopApp.iOS
 - Connected Services
 - ▶ References
 - ▶ Packages (3 updates)
 - Assets.xcassets
 - ▶ Properties
 - ▶ Rednerers
 - ▼ Resources

- americanexpresslogo.png
- americanexpresslogo@2x.png
- americanexpresslogo@3x.png
- Basket.png
- Basket@2x.png
- Basket@3x.png
- bestofYear.png
- bestofYear@2x.png
- bestofYear@3x.png
- bestShoes.png
- bestShoes@2x.png
- bestShoes@3x.png
- check.png
- check@2x.png
- check@3x.png
- clothing.jpg
- clothingCategory.png
- creditCardIcon.png
- electronics.jpeg
- electronicCategory.png
- garzy2.png
- garzy2@2x.png
- garzy2@3x.png
- grazy1.png
- grazy1@2x.png
- grazy1@3x.png
- Home.png
- Home@2x.png
- Home@3x.png
- ic_amex.png
- ic_amex@2x.png
- ic_amex@3x.png
- ic_amex_blue.png
- ic_amex_blue@2x.png
- ic_amex_blue@3x.png

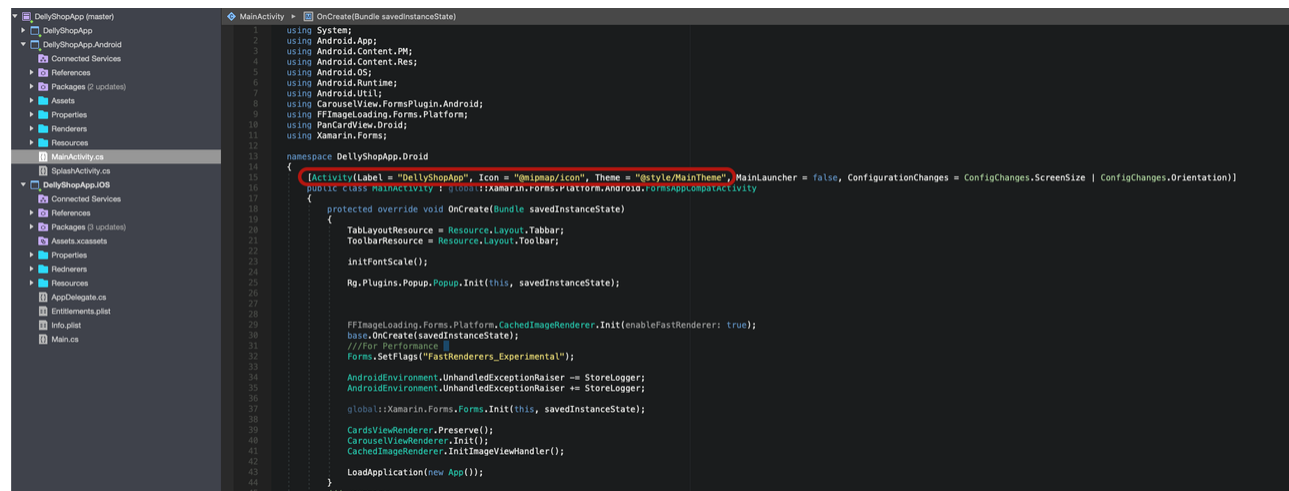


Replace the icons and the images which you want and add them as the same name dont change the names of your pictures

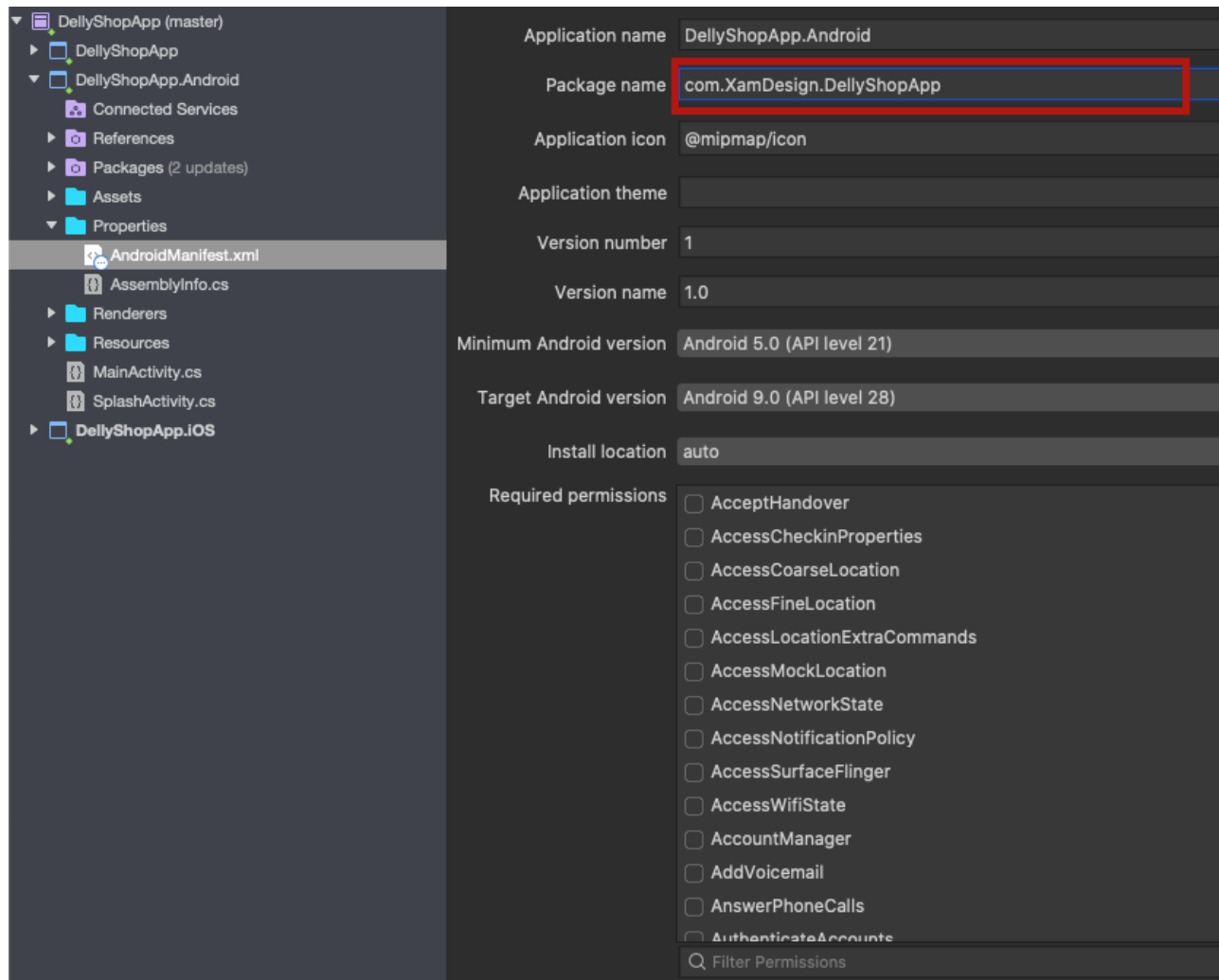
3-Change Application name & Package name

1-For Android app, to change application name, open **MainActivity.cs** file and change the variable bellow to your own Application name

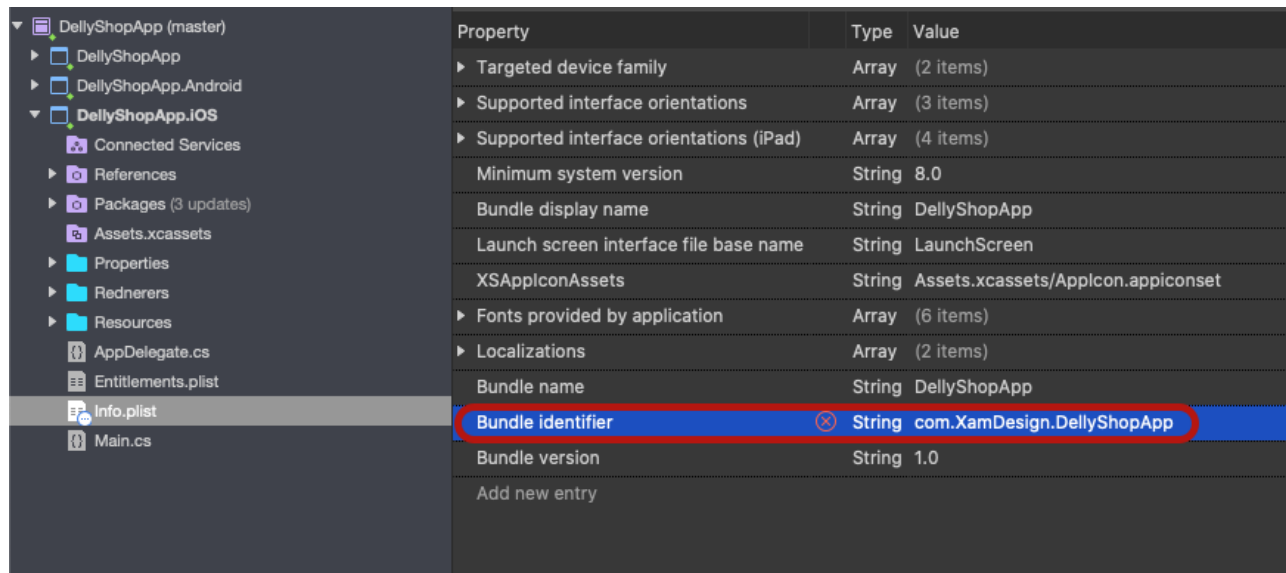
```
[Activity(Label = "DellyShopApp", Icon = "@mipmap/icon", Theme = "@style/MainTheme", MainLauncher = false, ConfigurationChanges = ConfigChanges.ScreenSize | ConfigChanges.Orientation)]
```



To change package name, double click on the project **Properties > Android Manifest** and change the package name value according to you.



2- For iOS application, double click on **Info.plist** , change the value of Bundle display name and Bundle name as you want for the application name, same for Bundle identifier as the package name.

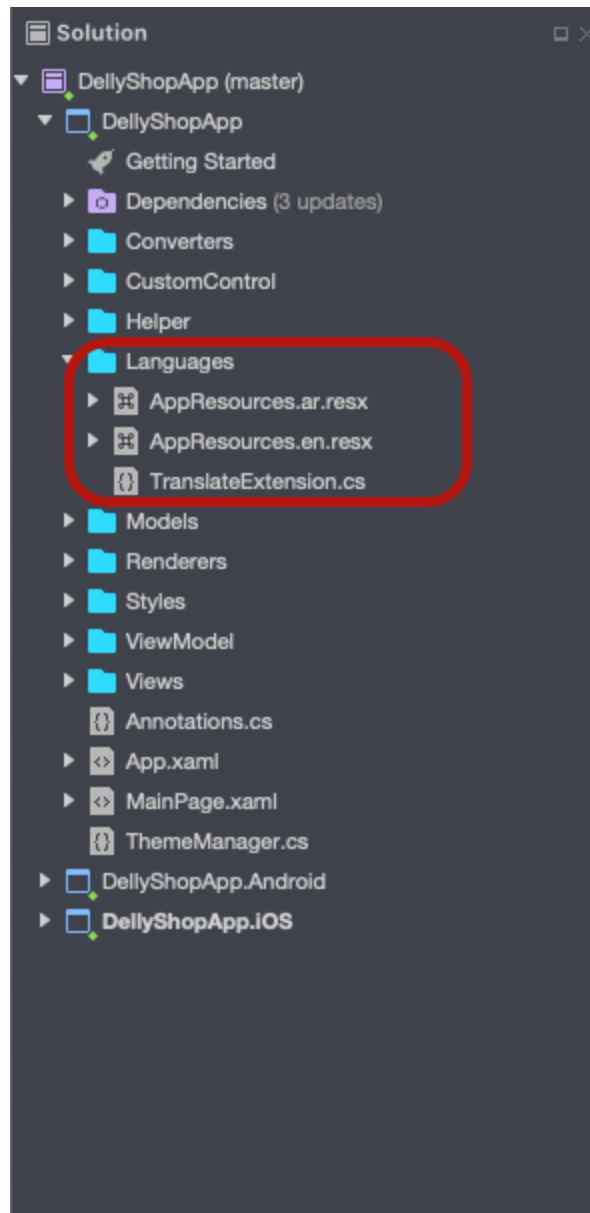


Change the red border fields as your needs

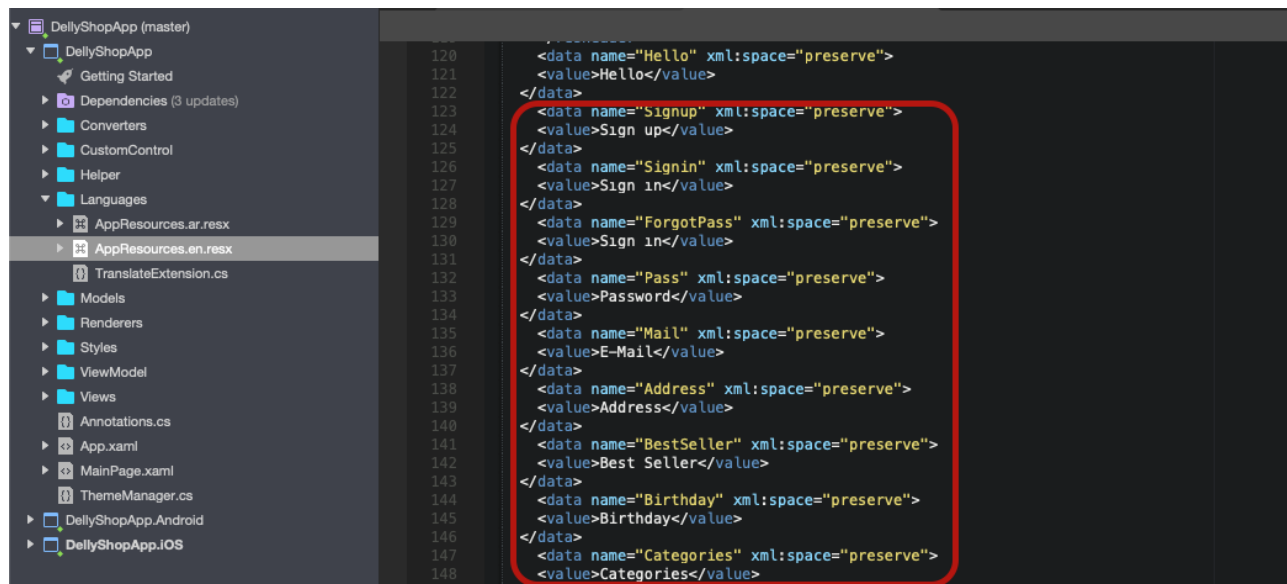
4-Change Language, Words and Translation

DellyShopApp Team made it for you easy to translate your own words in your app and change the labels as you want so lets start.

By default the folder **Languages** > contains all the English and Arabic words in the app



Your new translation should be added here with following format: AppResources.
<Your Language Code>.resx, simply open the new resx file and add your translation
in the Value column of the table



5-Change Splash Screen page ?

1- For Android, we create **SplashScreen** activity with specific theme for splash, you can change the splash icon by replacing the **splash_screen_light** icon under **drawable** folder.

Change the background to your own image which is located in the **Drawable** folder by renaming the red variable arrow on the image below to your own Image Name which should be added to your resource folder

DellyShopApp (master)

DellyShopApp

DellyShopApp.Android

Connected Services

References

Packages (2 updates)

Assets

Properties

Renderers

Resources

drawable

drawable-hdpi

drawable-ldpi

drawable-mdpi

drawable-xhdpi

drawable-xxhdpi

drawable-xxxhdpi

layout

mipmap-anydpi-v26

mipmap-hdpi

mipmap-mdpi

mipmap-xhdpi

mipmap-xxhdpi

mipmap-xxxhdpi

values

colors.xml

styles.xml

values-night

AboutResources.txt

Resource.designer.cs

MainActivity.cs

SplashActivity.cs

DellyShopApp.iOS

1<?xml version="1.0" encoding="utf-8" ?>

2<resources>

3

4<style name="MainTheme" parent="MainTheme.Base">

5<item name="android:navigationBarColor">#EE5527</item>

6</style>

7<style name="MainTheme.Base" parent="Theme.AppCompat.Light.DarkActionBar">

8<item name="windowNoTitle">true</item>

9<item name="windowActionBar">false</item>

10<item name="colorPrimary">#EE5527</item>

11<item name="colorPrimaryDark">#EE5527</item>

12<item name="colorAccent">#EE5527</item>

13<item name="windowActionModeOverlay">true</item>

14<item name="android:datepickerDialogTheme">@style/AppCompatDialogStyle</item>

15

16</style>

17<style name="MyTheme.Splash" parent="Theme.AppCompat.Light.NoActionBar">

18<item name="android:windowBackground">@drawable/splash_screen_light</item>

19<item name="android:windowNoTitle">true</item>

20<item name="android:navigationBarColor">#EE5527</item>

21<item name="android:windowFullscreen">true</item>

22<item name="android:windowContentOverlay">@null</item>

23<item name="android:windowActionBar">true</item>

24</style>

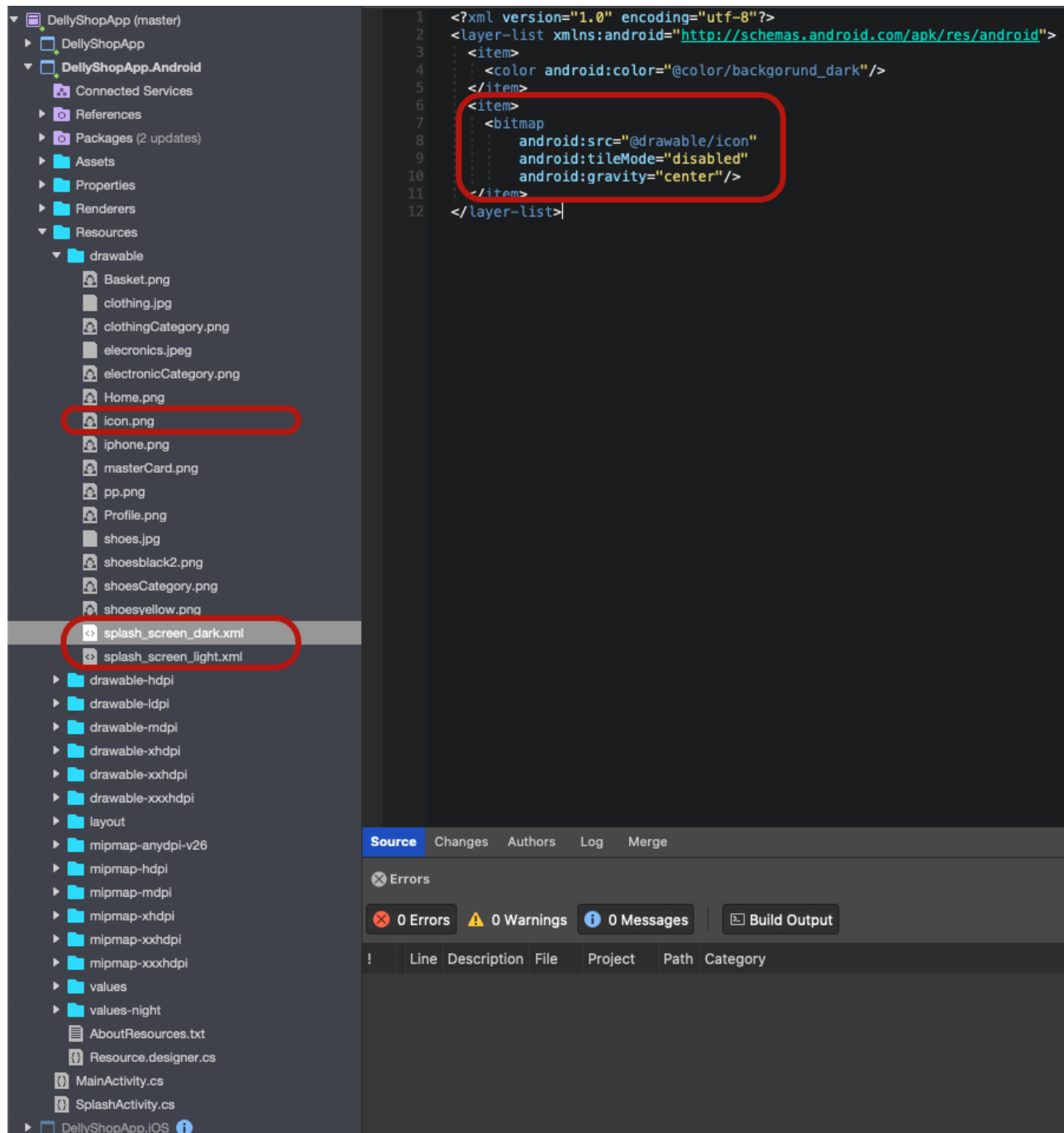
25<style name="AppCompatDialogStyle" parent="Theme.AppCompat.Light.Dialog">

26<item name="colorAccent">#EE5527</item>

27</style>

28</resources>

29



Note: Every image must be added on all the Drawable folders to let it work on all different screen sizes

2- For iOS, we implement launch screen with **storyboard**, in the **DellyShopApp.iOS** project > **Resources** folder you will see the **LaunchScreen.storyboard**. To customise the story design, you need a Mac device, refer to follow instruction to change launcher image on iOS app:

Set Up Device for Development

This article will discuss how to setup an Android device and connect it to a computer so that the device may be used to run and debug Xamarin.Android applications. By now, you've probably seen your great new application running on the Android emulator, and want to see it running on your shiny Android device. Here are the steps involved with connecting a device to a computer for debugging:

1. **Enable Debugging on the Device** - By default, it will not be possible to debug applications on a Android device.
2. **Install USB Drivers** - This step is not necessary for OS X computers. Windows computers may require the installation of USB drivers.
3. **Connect the Device to the Computer** - The final step involves connecting the device to the computer by either USB or WiFi.

Enable Debugging on the Device

It is possible to use any Android device to test an Android application. However the device must be properly configured before debugging can occur. The steps involved are slightly different, depending on the version of Android running on the device.

Android 4.0 to Android 4.1

For Android 4.0.x to Android 4.1.x, debugging is enabled by following these steps:

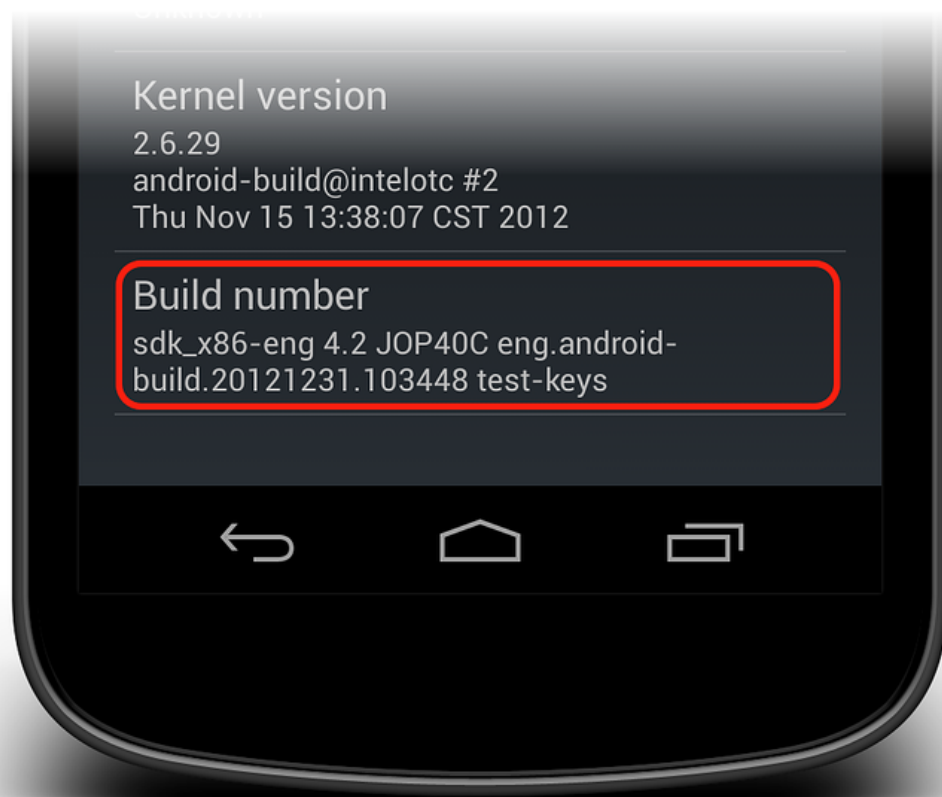
1. Go to the **Settings** screen.
2. Select **Developer options** .
3. Check off the **USB debugging** option.

This screenshot shows the Developer options screen on a device running Android 4.0.3:

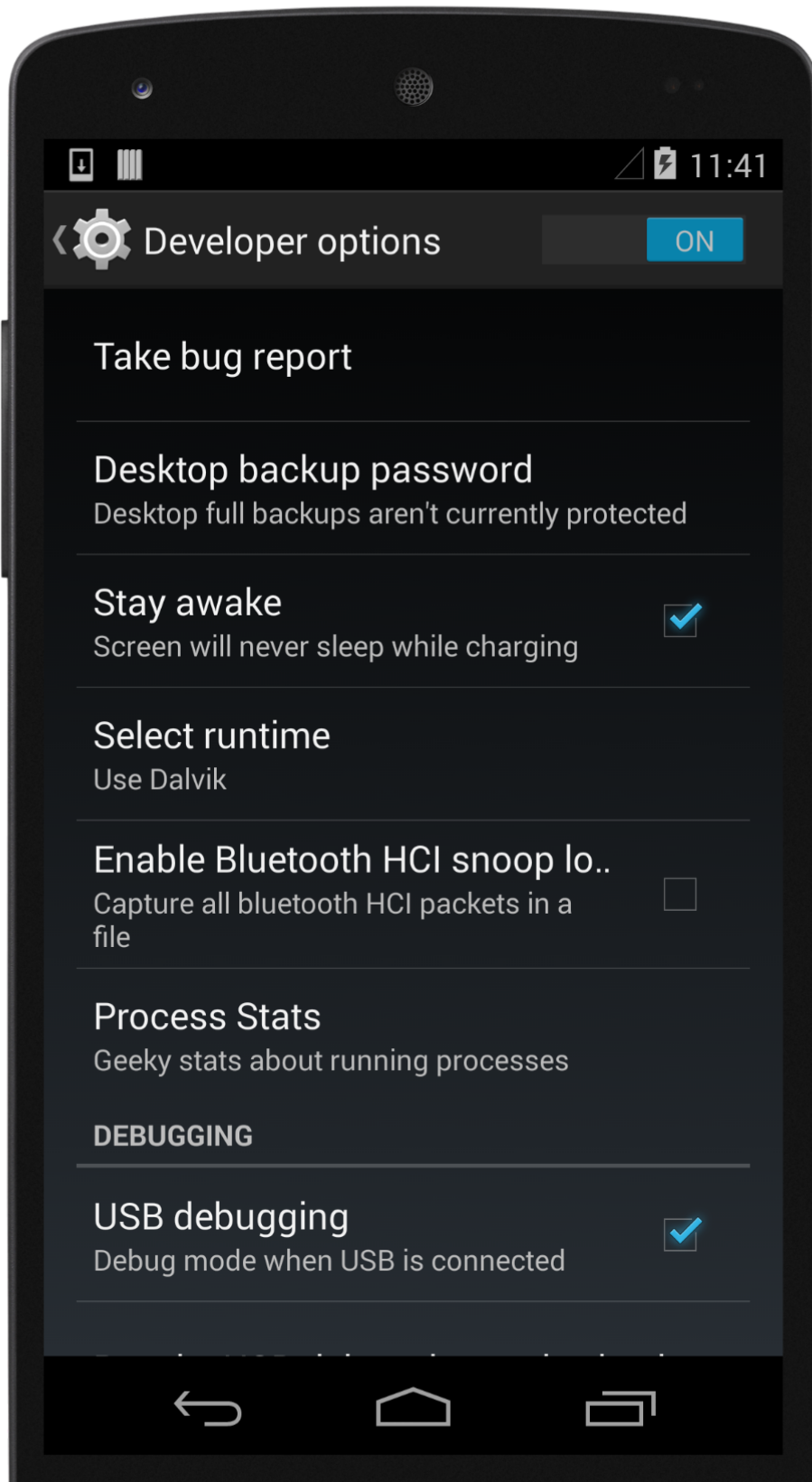


Android 4.2 and higher till 7.2

Starting in Android 4.2 and higher, the Developer options is hidden by default. To make it available, go to Settings > About phone, and tap the Build number item seven times to reveal the Developer Options tab:



Once the **Developer Options** tab is available under **Settings** > System, open it to reveal developer settings:



11:41



Developer options

ON

Take bug report

Desktop backup password

Desktop full backups aren't currently protected

Stay awake

Screen will never sleep while charging



Select runtime

Use Dalvik

Enable Bluetooth HCI snoop log

Capture all bluetooth HCI packets in a file



Process Stats

Geeky stats about running processes

DEBUGGING

USB debugging

Debug mode when USB is connected





This is the place to enable developer options such as USB debugging and stay awake mode.

Install USB Drivers

This step is not necessary for OS X. Just connect the device to the the Mac with a USB cable.

It may be necessary to install some extra drivers before a Windows computer will recognize an Android device connected by USB.

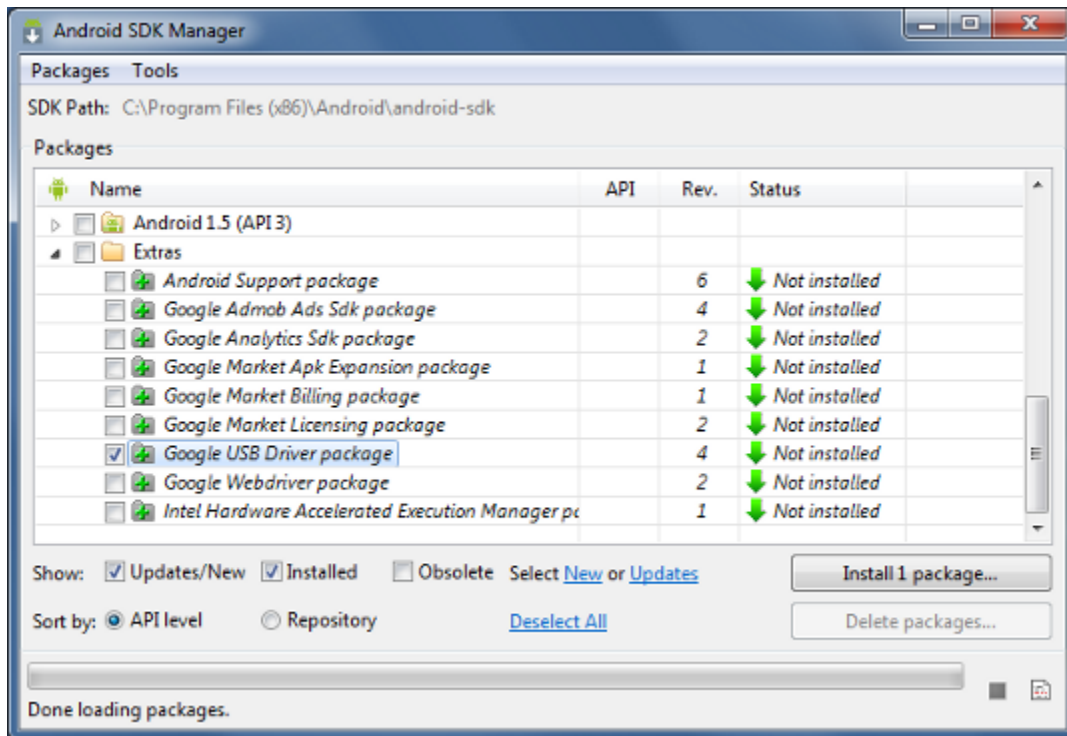
Note: These are the steps to set up a Google Nexus device and are provided as a reference. Steps for your specific device may vary, but will follow a similar pattern. Search the internet for your device if you have trouble.

Download the USB Drivers

Google Nexus devices (with the exception of the Galaxy Nexus) require the Google USB Driver. The driver for the Galaxy Nexus is [distributed by Samsung](#).

All other Android devices should use the [USB driver from their respective manufacturer](#).

Install the **Google USB Driver** package by starting the Android SDK Manager, and expanding the **Extras** folder, as can be seen in the follow screenshot:



Check the **Google USB Driver** box, and click the **Install** button.

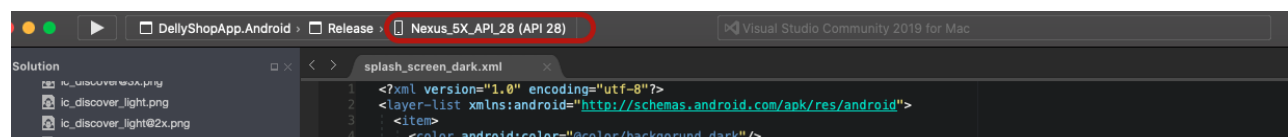
Connect the Device to the Computer

The final step is to connect the device to the computer. There are two ways to do so:

- **USB cable** - This is the easiest and most common way. Just plug the USB cable into the device and then into the computer.
- **WiFi** - It is possible to connect an Android device to a computer without using a USB cable, over WiFi. This technique requires a bit more effort but could be useful when there is no USB cable or the device is too far away for a USB cable. Connecting via WiFi will be covered [Here](#).

Now Your device will appear in the Debug section click start and the application will be deployed to your own device.

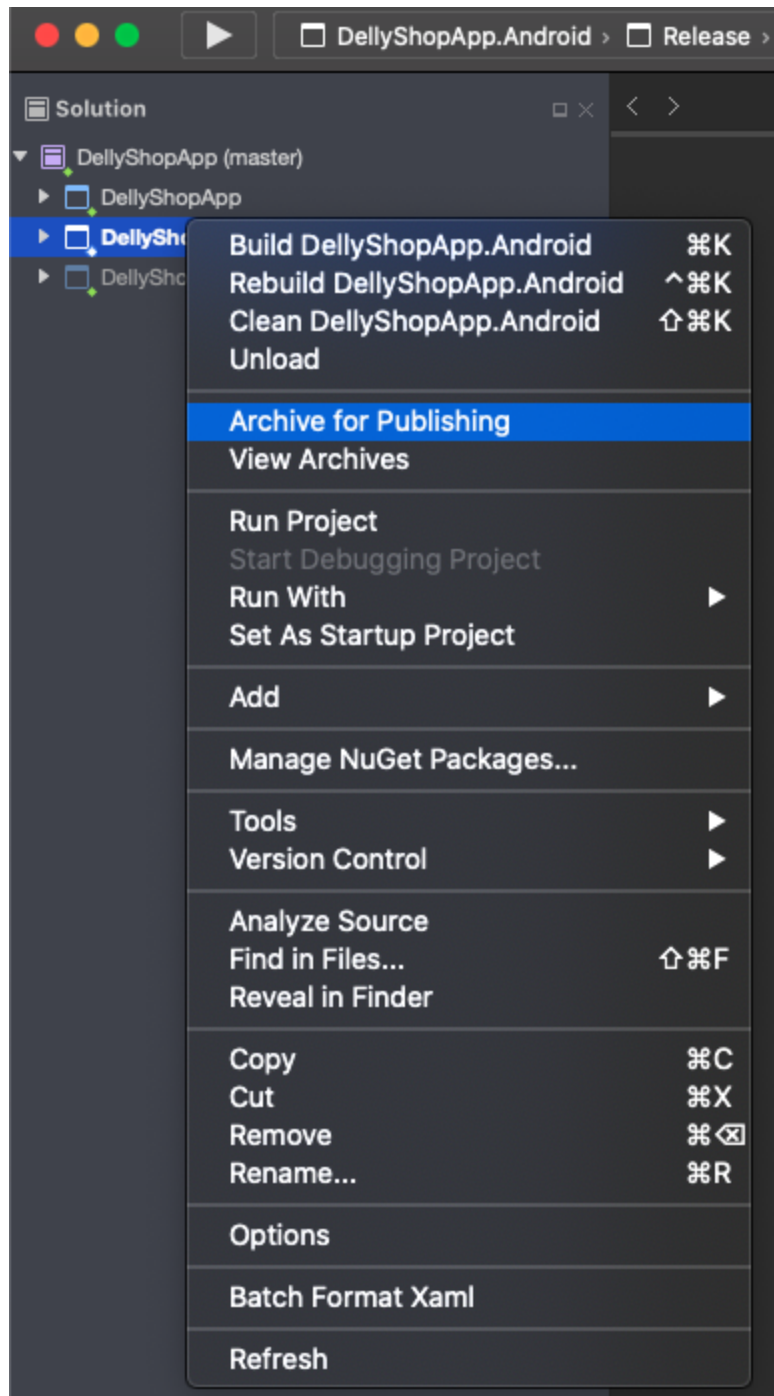
Now you can start your own test



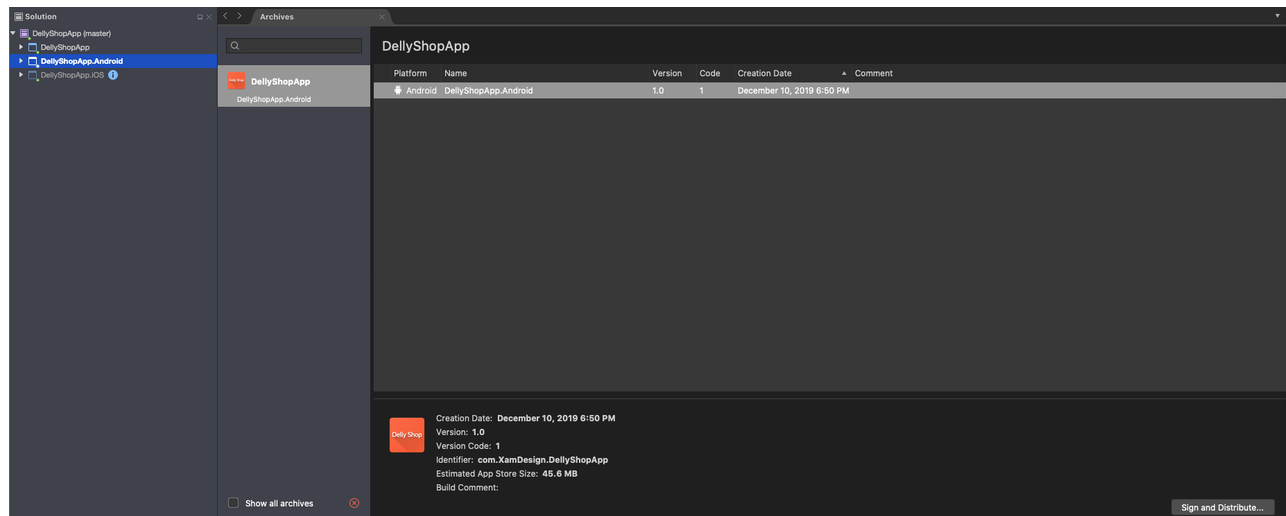
Android / Publishing Application

1 - Archive for Publishing

To begin the publishing process, right-click the project in Solution Explorer and select the Archive... context menu item:



Archive... launches the Archive Manager and begins the process of archiving the App bundle as shown in this screenshot:



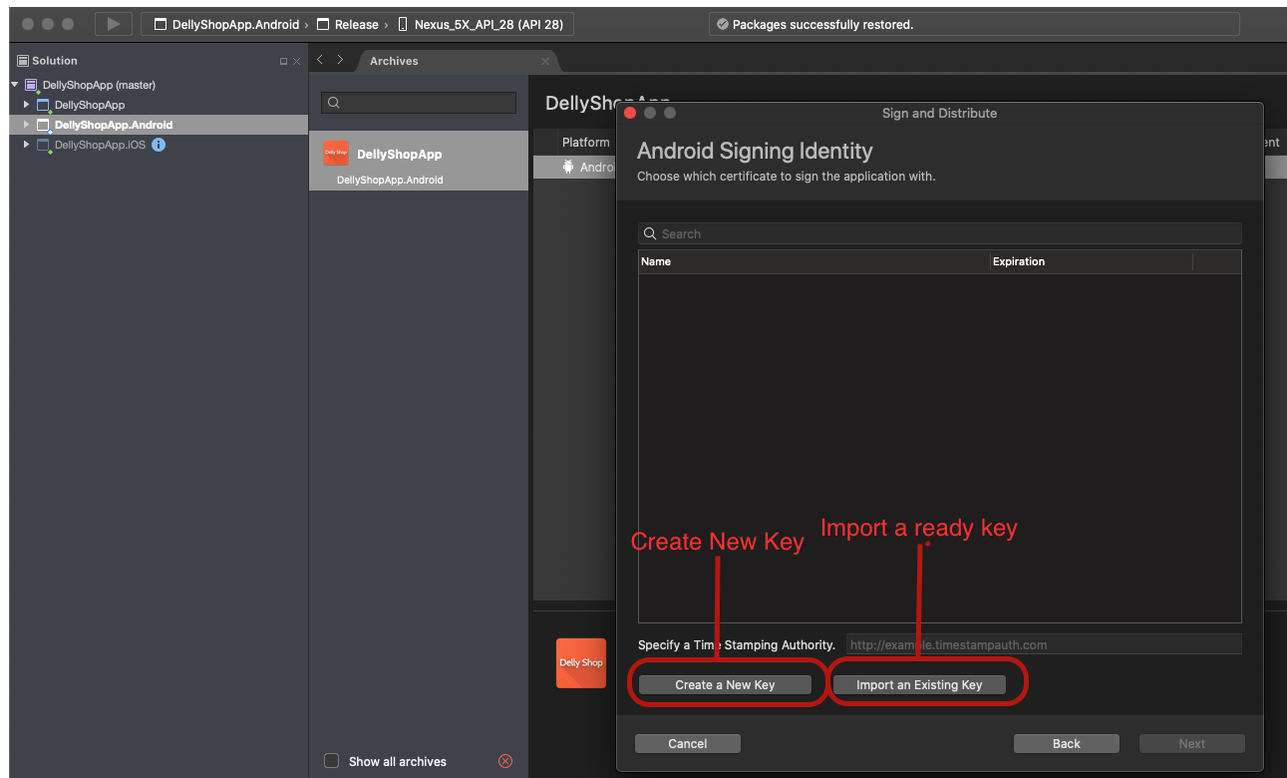
2 - Distribution:

When an archived version of the application is ready to publish, select the archive in the Archive Manager and click the Distribute... button:

The Distribution Channel dialog presented two choices for distribution. Select Ad-Hoc:

Create a New Certificate

After Ad-Hoc is selected, Visual Studio opens the Signing Identity page of the dialog as shown in the next screenshot. To publish the .APK, it must first be signed with a signing key (also referred to as a certificate). An existing certificate can be used by clicking the Import button and then proceeding to Sign the APK. Otherwise, click the click the + button to create a new certificate:



The Create Android Key Store dialog is displayed; use this dialog to create a new signing certificate that can use for signing Android applications. Enter the required information (outlined in red) as shown in this dialog:

The resulting keystore resides in the following location: (Save Them)

(C:\Users\USERNAME\AppData\Local\Xamarin\Mono for Android\alias\alias.keystore)


For example, the above steps might create a new signing key in the following location:

(C:\Users\USERNAME\AppData\Local\Xamarin\Mono for Android\chimp\chimp.keystore)

iOS / Publishing Application

It's a little more complicated on iOS app publishing. Make sure you have read and understand Apple's App Store Guidelines. You can refer to following instructions to proceed upload your app:

https://medium.com/@the_manifest/how-to-publish-your-app-on-apples-app-store-in-2018-f76f22a5c33a

 How to Publish Your App on Apple's App Store in 2018 • medium.com

Frequently Asked Questions (FAQ)

What Minimum & Maximum Android and iOS Target this Application supports?

- Minimum: Android Target is V5.0, iOS target is 9.0
- Maximum: Android Target is Version v10.0 and works fine on higher versions. iOS with latest version 13.2

Is this Application Native or Hybrid ?

- This is full Shared code 95% and 5% platform specific code used on Import images section and another section

How can i modify the style?

- You can change it from Settings class in your main solution.
- You can read the Customization section on the doc file

Can you install this application for me ?

- Yes we can , By buying Extended license , We install for free on each new update.
- Or by paying 20\$ for each new install

How i will know if there is new update for this Application ?

- Once there is an new update you will get an email send by us with the new update url so you can download and update your app as you like.

How i can have a custom work for this application ?

- Once You decide to customize your application an change many things or add something new, you can send us a request via our [Codecanyon](#) account or directly to email address XamAppDesign@gmail.com
-