

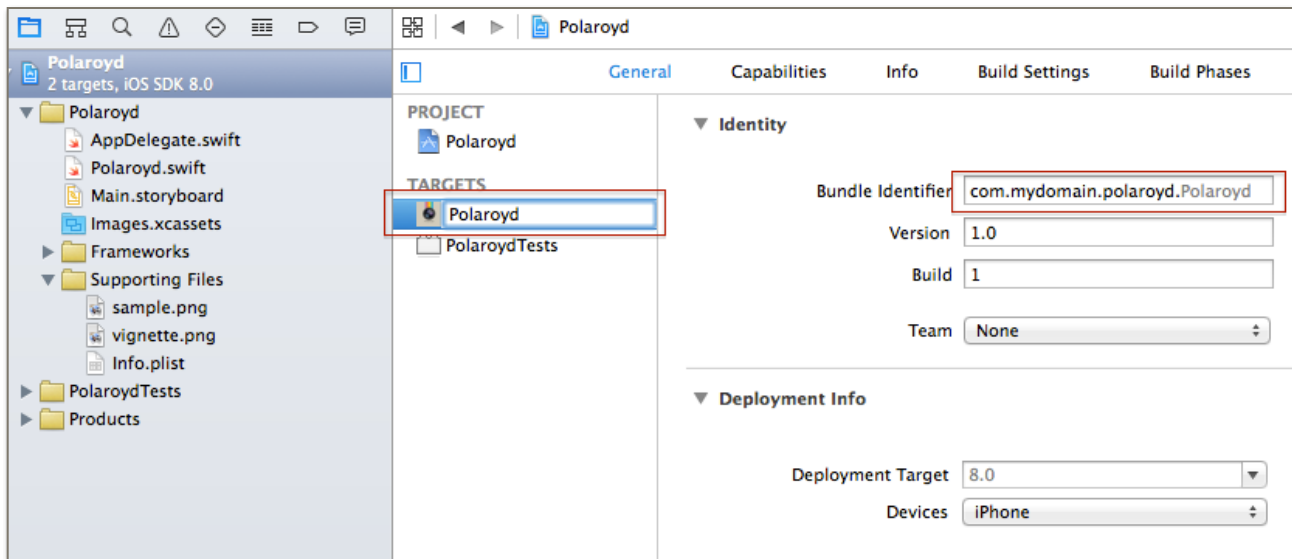
# Polaroyd

## User Guide

Thanks for purchasing **Polaroyd** Full iOS Vintage Photo Editor App Template from CodeCanyon, we really appreciate your support and we're sure you'll have so much fun with this template!

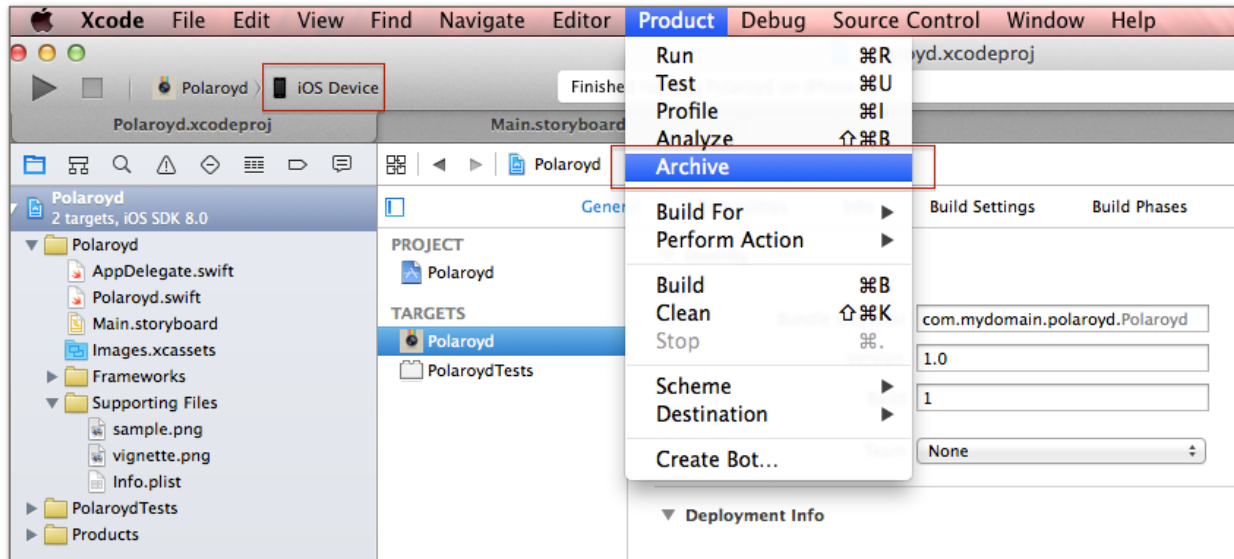
### - First Setup -

- **Bundle Identifier & App name:** If you want to use the demo project included into the package, you can do so. Once you have generated a new **App ID** and **Distribution Provisioning** file from the **Apple Developer portal** (in order for you to be able to upload the binary of your app), you must change the **Bundle Identifier** accordingly to the one you generated in your provisioning profile. Also, you must change the App's name under **TARGETS** by double clicking on **Polaroyd** and renaming it as you want, so you'll see your own app's name underneath the app's icon (see below):



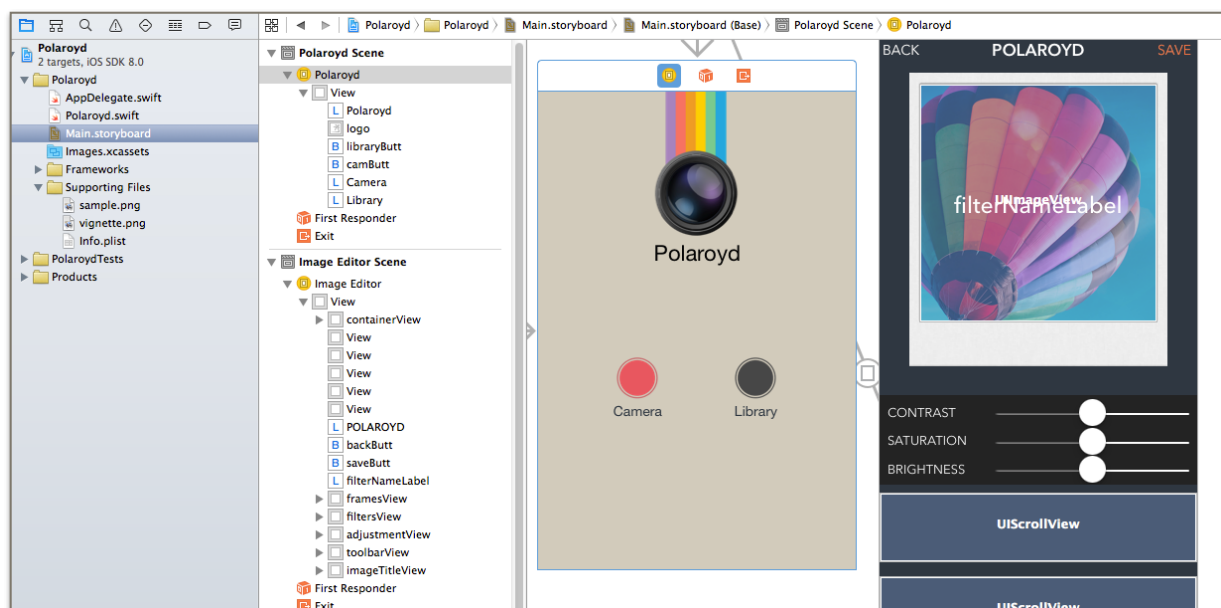
**NOTE:** do not change the Polaroyd folder's name or Project's name, or else XCode 6 won't recognize the app and you won't be able to test it on the iOS Simulator nor Archive it for AppStore publishing.

- **Archive the App:** To archive the app and be able to upload it to the iTunes AppStore, select **iOS Device** on the device list and then click on **Product -> Archive** to launch the **Organizer** and archive the app (see below):



## - Customization -

- **User Interface:** If you want to edit the User Interface of **Polaroyd**, you can do so by the Main Storyboard file called **Main.storyboard**, as shown below. You can easily change colors and images by selecting the elements you need to change and edit them on the XCode right panel.
- **Edit Filters:** If you want to edit some Filter settings in order to change the ones we've created, first of all take a look at the Apple Core Image



CoreImage Filter Reference page here: <https://developer.apple.com/library/ios/documentation/graphicsimaging/reference/CoreImageFilterReference/index.html>

Then just go into the `ImageEdit.swift` file, find

```
func filterButtTapped(button: UIButton) {
```

method and edit the values of the **case** sentences contained into such method. Here's an example on **case 9**:

```
case 9: // Green
    var color:UIColor = UIColor(red: 99/255.0, green: 218/255.0, blue:
90/255.0, alpha: 1.0)
    filter!.setValue(CIColor(color: color), forKey: kCIInputColorKey)
    filter!.setValue(0.5, forKey: kCIInputIntensityKey)
    break
```

In this case, you can change the numbers in red bold color, the first ones are the RGBA values for colours, keep in mind that **red**, **green** and **blue** values go from 0 to 255, and **alpha** value goes from 0.0 to 1.0. Same thing for **intensity** value on the 3rd line of code, it goes from 0.0 to 1.0.

Play with values of the that **case 9** sentence and test the app to customise filters as you wish.

•**Add Filters:** If you want to add some filters, you can copy some of the existing ones and just edit them, since we've used almost all the CIFilters that work on iOS. Here's a step-by-step example:

-Right above the `class ImageEditor: UIViewController` class declaration, you'll see a couple of Arrays that store CIFilters original names and custom Filter names.

Place this string right below the last Array's item in `filtersArray: Array`  
`"CIColorMonochrome", //12`

Place this string right below the last item in `filterNamesArray: Array`  
`"Brown", //12`

So **Brown** filter will be our 12th new filter, based on **CIColorMonochrome** Core Image Filter (*aka CIFilter*).

-Copy the whole **case 9** (see below):

```
case 9: // Green
```

```

        var color:UIColor = UIColor(red: 99/255.0, green: 218/255.0, blue:
90/255.0, alpha: 1.0)
        filter!.setValue(CIColor(color: color), forKey: kCIInputColorKey)
        filter!.setValue(0.5, forKey: kCIInputIntensityKey)
        break

```

And paste it right below the **case 11**.

-You can now edit the RGBA and intensity values and create your Brown filter. The brown color RGB values are **171-113-39**, and let's set its intensity as **0.9**.

So here's the complete new case:

```

case 18: // Brown
    var color:UIColor = UIColor(red: 171/255.0, green: 113/255.0, blue:
39/255.0, alpha: 1.0)
    filter.setValue(CIColor(color: color), forKey: kCIInputColorKey)
    filter.setValue(0.9, forKey: kCIInputIntensityKey)
    break

```

Your new Brown filter is now ready to be used in your Polaroyd app!

The only thing you have to do is create its proper icon, go into *Images.xcassets/Filter icons*, add the **f11** Image Set and add your 2x and 3x png icons for your new filter.

•**AdMob setup:** In order for you to display your own AdMob banners you must create an AD UNIT ID on [www.apps.admob.com](http://www.apps.admob.com). Once you've got it, replace the red string below with your **AD UNIT ID** (on the top of **Home.swift** file):

```

let ADMOB_UNIT_ID = "ca-app-pub-9733347540588953/7805958028"

```

•**Apple Review process:** Sometimes it may happen that an app gets rejected once by Apple reviewers because of ads (iAd or AdMob doesn't matter), that's an issue we encountered too and we can't still understand why since we then re-submitted an app without any change and it got approved.

Anyway, check out the screenshot below, you'll get it after clicking on Submit For Review button on iTunes Connect, check the options shown below and everything should go smoothly.

Advertising Identifier

Does this app use the Advertising Identifier (IDFA)?

☒ Yes

☐ No

The [Advertising Identifier \(IDFA\)](#) is a unique ID for each iOS device and is the only way to offer targeted ads. Users can choose to limit ad targeting on their iOS device.

If your app is using the Advertising Identifier, check your code—including any third-party code—before you submit it to make sure that your app uses the Advertising Identifier only for the purposes listed below and respects the Limit Ad Tracking setting. If you include third-party code in your app, you are responsible for the behavior of such code, so be sure to check with your third-party provider to confirm compliance with the usage limitations of the Advertising Identifier and the Limit Ad Tracking setting.

This app uses the Advertising Identifier to (select all that apply):

☒ Serve advertisements within the app

☐ Attribute this app installation to a previously served advertisement

☐ Attribute this app installation to a previously served advertisement

If you think you have another acceptable use for the Advertising Identifier, [contact us](#).

Limit Ad Tracking setting in iOS

☒

, Anne Johnson, confirm that this app, and any third party that interfaces with this app, uses the Advertising Identifier checks and honors a user's Limit Ad Tracking setting in iOS and, when it is enabled by a user, this app does not use Advertising Identifier, and any information obtained through the use of the Advertising Identifier, in any way other than for "Limited Advertising Purposes" as defined in the [iOS Developer Program License Agreement](#).

Have fun with [Polaroyd](#), and don't forget to rate it on your Downloads page!



[fvimagination.com](http://fvimagination.com)