



AdWhirl Open Source SDK
Setup Instructions
for iPhone SDK 2.2.1

AdWhirl Open Source SDK Setup Instructions for iPhone SDK 2.2.1

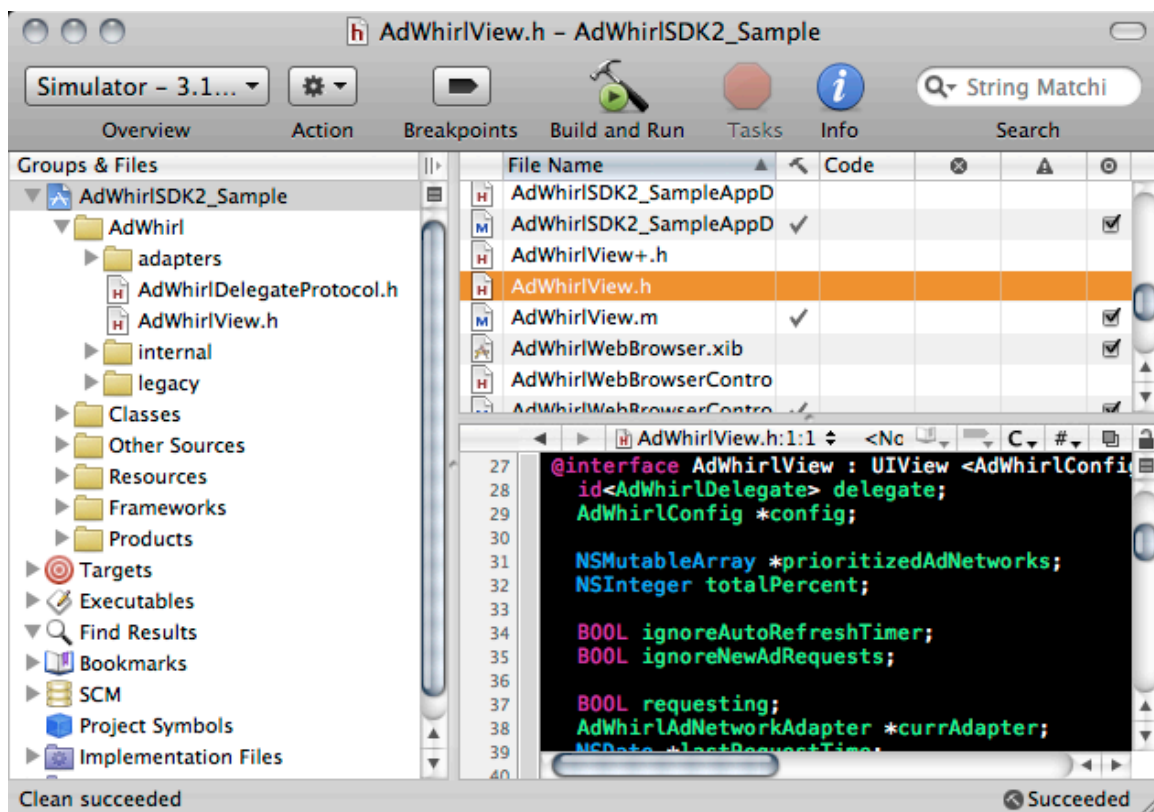
The AdWhirl Open Source SDK contains the code to enable your iPhone application to display ads from different ad networks.

Setting up the AdWhirl SDK for iPhone SDK 2.2.1

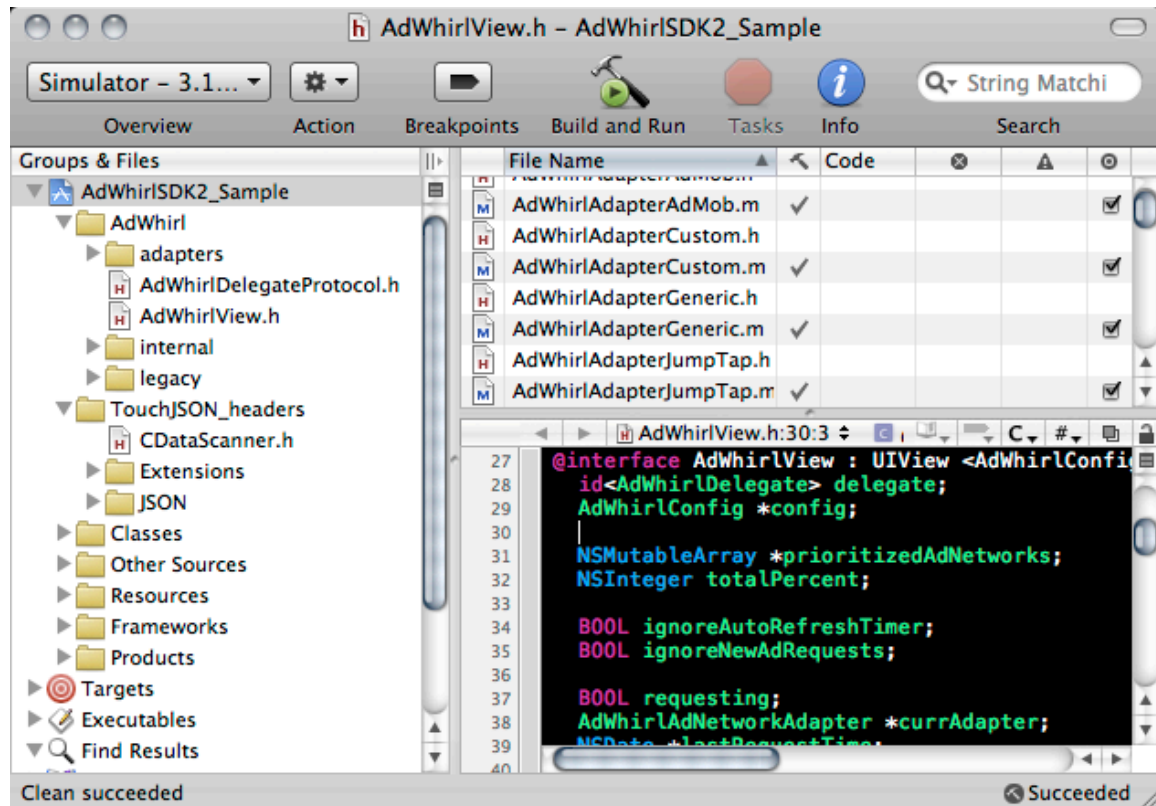
The instructions below assume that you are familiar with Xcode, and understand how to change build settings and add frameworks to your Xcode projects. They also assume that you are developing for iPhone SDK 2.2.1.

Note: When adding files to your Xcode project, make sure "Copy items into destination group's folder (if needed)" is checked and use "Recursively create groups for any added folders."

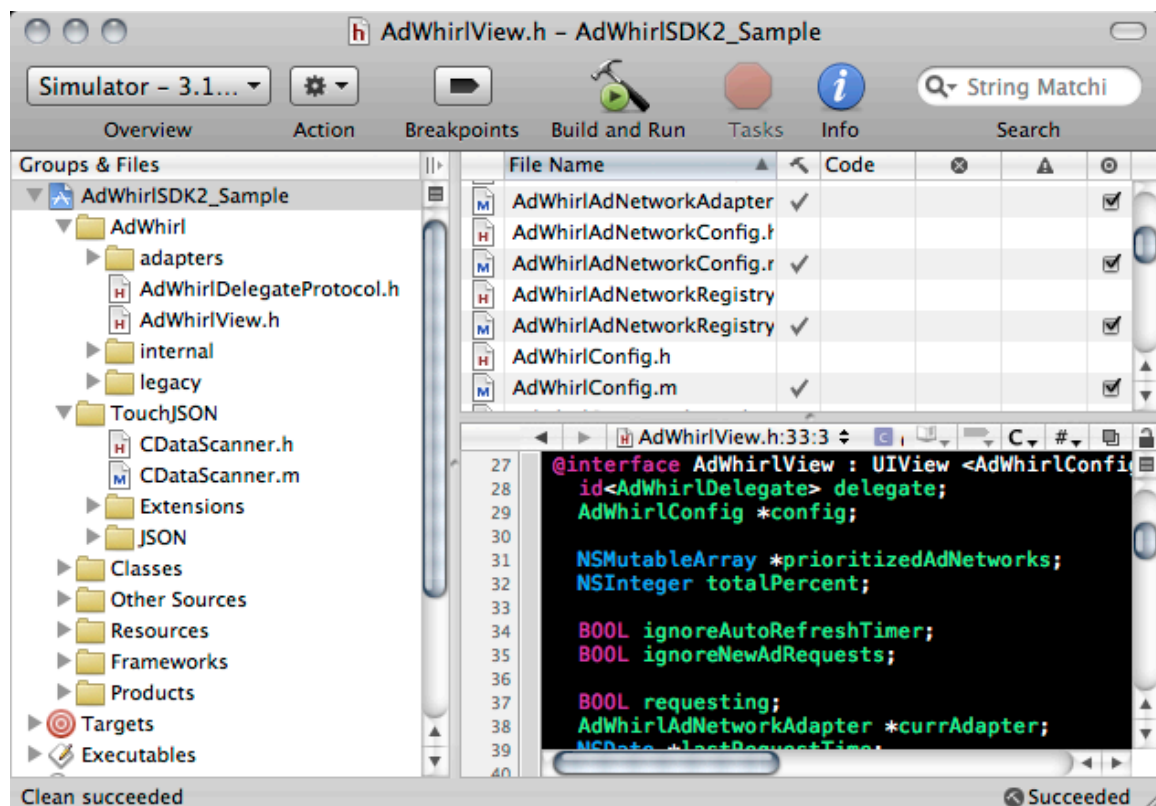
1. Add the AdWhirl folder from the AdWhirl SDK into your project. In the AdWhirl folder, you'll find two files (AdWhirlDelegateProtocol.h, AdWhirlView.h) and three folders (adapters, internal, legacy).



2. Add TouchJSON into your project. If you plan to include the AdMob SDK, drag the TouchJSON_headers directory into your project.



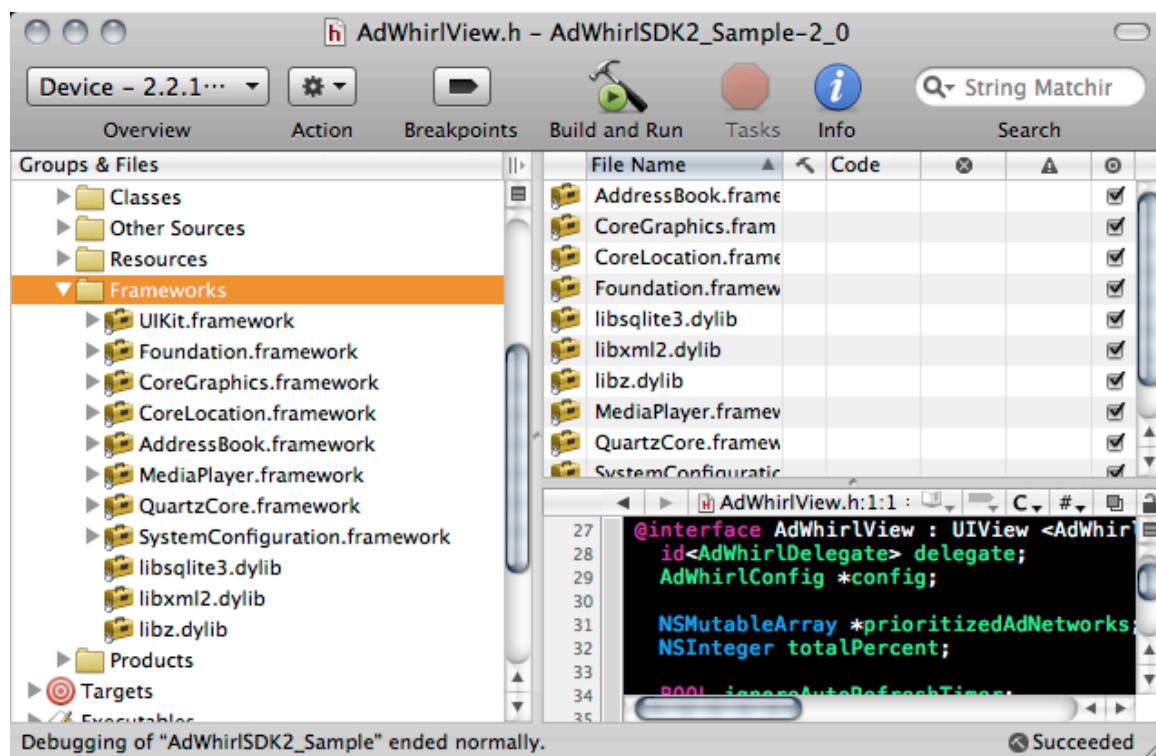
Otherwise, drag the TouchJSON directory into your project.



3. Add the supporting frameworks required by all of the supported ad networks.

- CoreLocation.framework
- AddressBook.framework
- MediaPlayer.framework
- QuartzCore.framework
- SystemConfiguration.framework
- libsqlite3.dylib
- libxml2.dylib
- libz.dylib

The frameworks do not affect the size of your executable unless they are used by ad network libraries that you integrate.

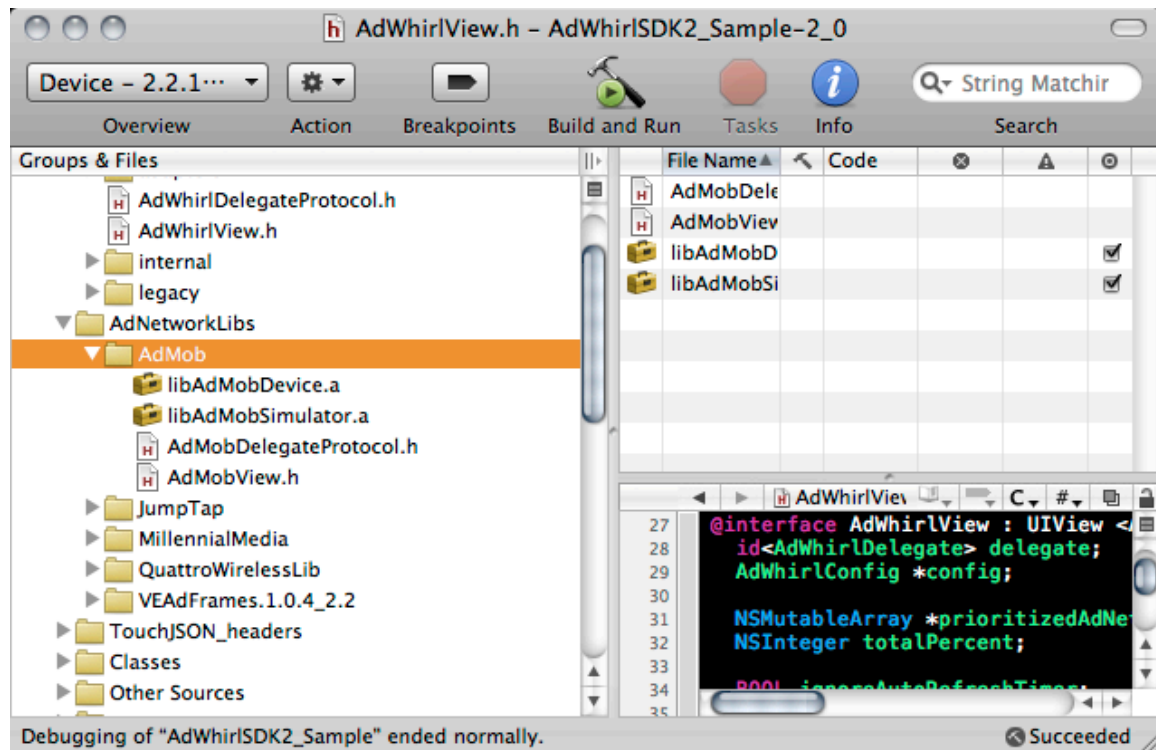


Note: UIKit.framework, Foundation.framework, and CoreGraphics.framework are included by default.

4. Add the ad network libraries. All of the ad network libraries are optional. You do not have to integrate an ad network if you don't want to run the ads for that network, or don't have access to the SDK.

a. AdMob

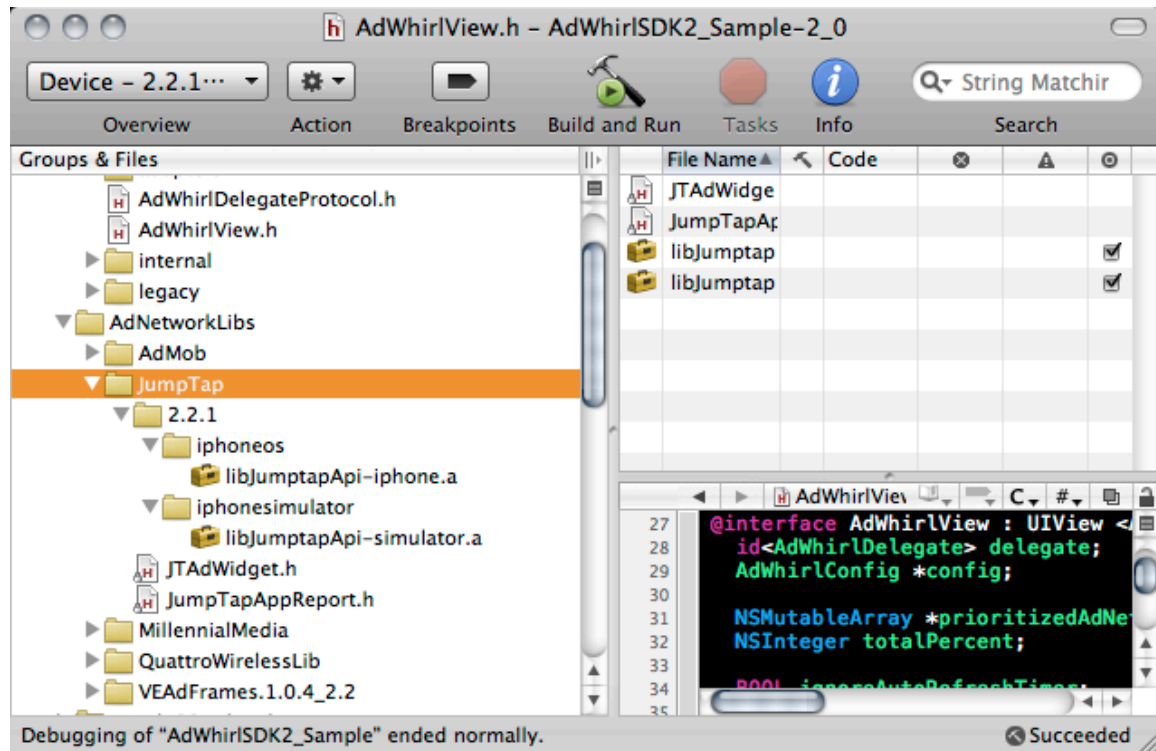
- i. Download the AdMob SDK from <http://www.admob.com>.
- ii. Drag the AdMob folder into your Xcode project (containing AdMobDelegateProtocol.h, AdMobView.h, libAdMobSimulator.a, libAdMobDevice.a).



b. JumpTap

- i. Download the JumpTap SDK from <http://www.jumptap.com>.
- ii. Drag the JumpTap folder into your Xcode project (containing the folders 2.2.1 and 3.0, and the files JTAdWidget.h and JumpTapAppReport.h).
- iii. Remove the 3.0 folder.

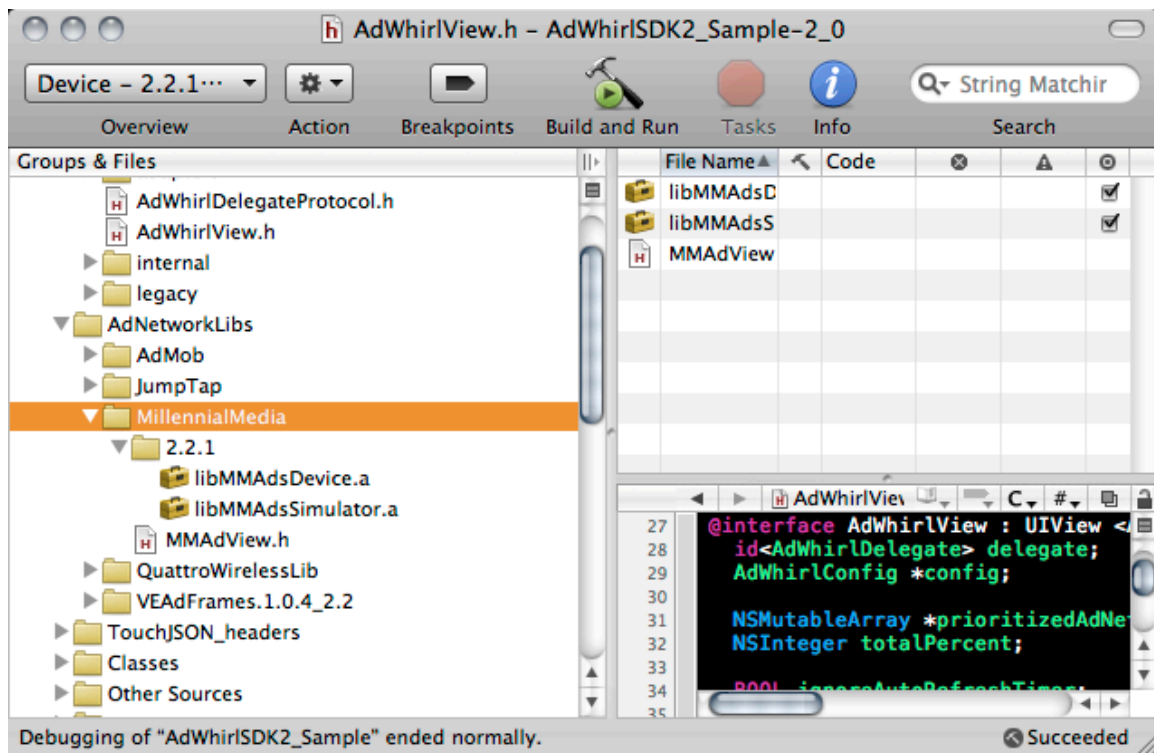
- iv. To avoid build errors, rename `libJumptapApi.a` in the `iphonios` and `iphonesimulator` directories to include the architecture in the file names, for example, `libJumptapApi-iphone.a` and `libJumptapApi-simulator.a`.



c. Millennial Media

- i. Download the Millennial Media SDK from <http://www.millennialmedia.com>.
- ii. Drag the `MillennialMedia` folder into your Xcode project (containing the folders `2.2.1` and `3.0`, and the file `MMAView.h`).

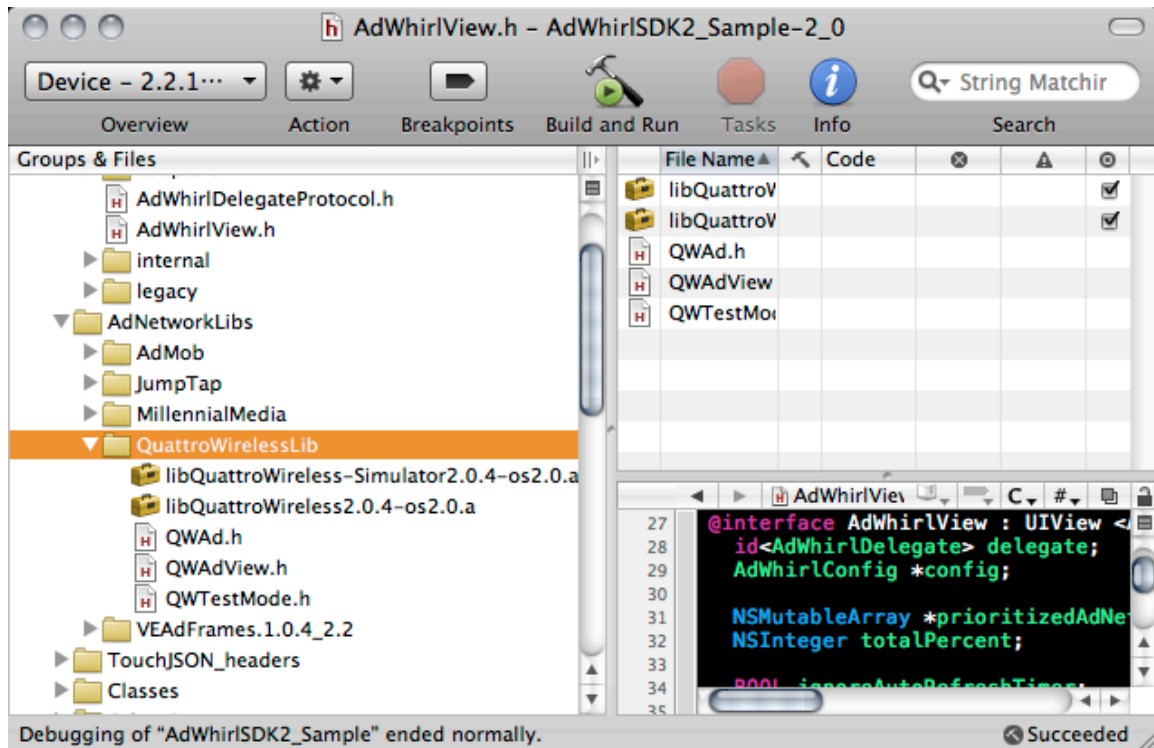
- iii. Remove the 3.0 folder.



d. Quattro Wireless

- i. Download the Quattro Wireless SDK from <http://www.quattrowireless.com>. Be sure to download the version for OS 2.2.1.
- ii. Drag the QuattroWirelessLib folder into your Xcode project (containing libQuattroWireless-Simulator2.0.4-os2.0.a, libQuattroWireless2.0.4-os2.0.a, QWTestMode.h, QWAd.h, and QWAdView.h. Exact names of .a files may be different depending on the version you download).

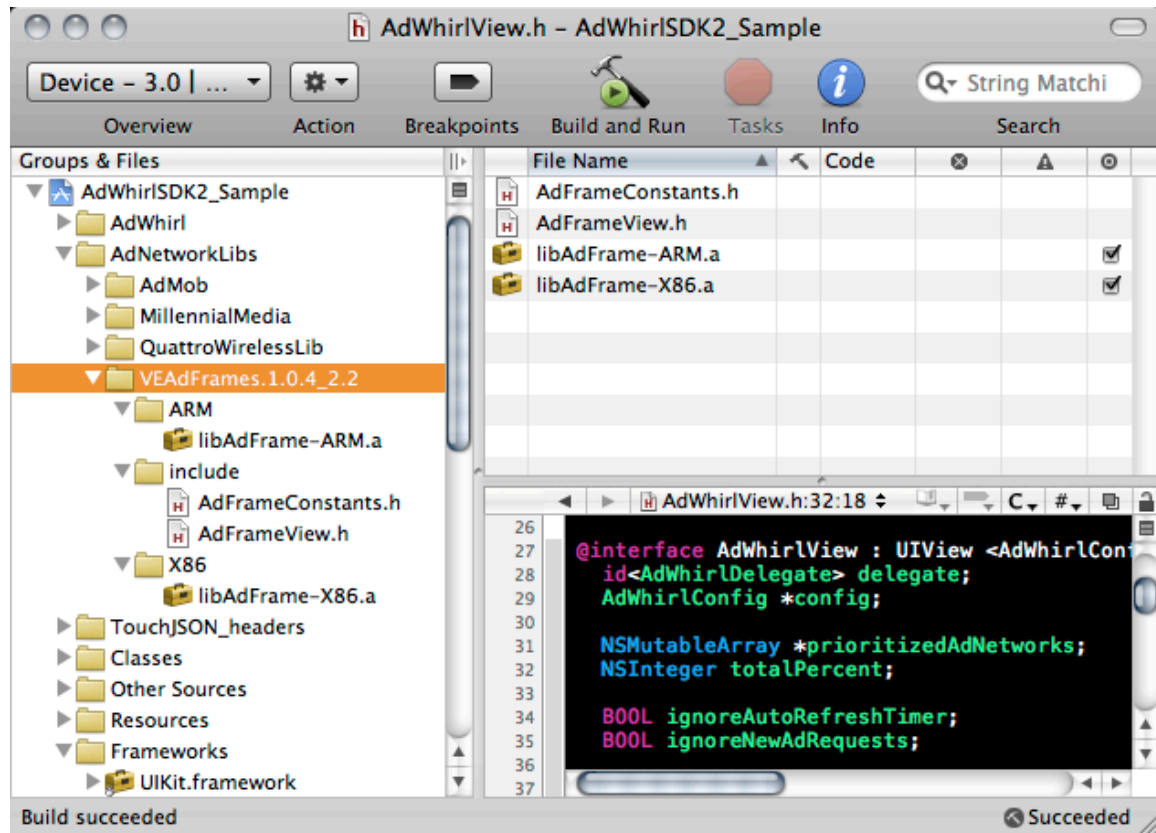
- iii. Go to Project -> Project Settings. Under the Build tab, include the `-ObjC` flag in the Other Linker Flags setting for all configurations (make sure the Configuration pulldown menu reads "All Configurations").



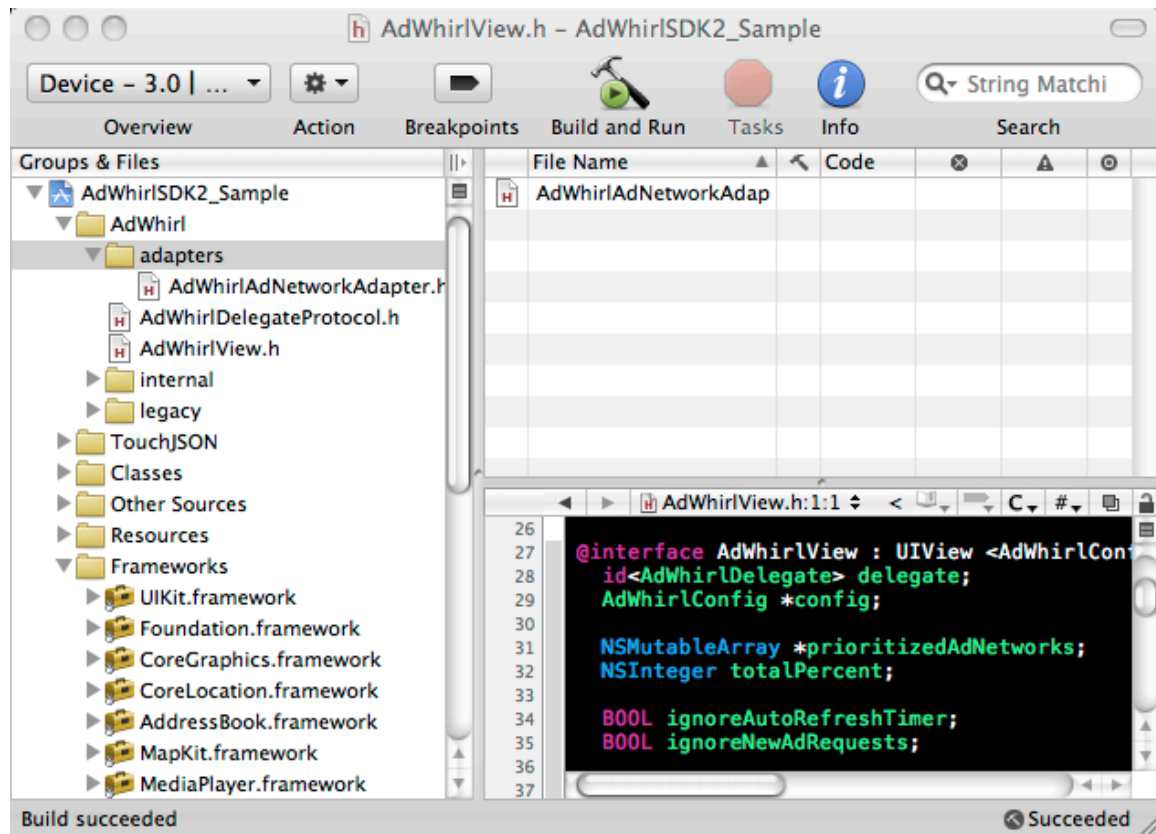
e. VideoEgg

- i. Download the VideoEgg SDK from <http://www.videoegg.com/>.
- ii. Drag the appropriate `VEAdFrames` folder under the `lib` folder into your project (containing the folders `ARM`, `x86`, and `include`).

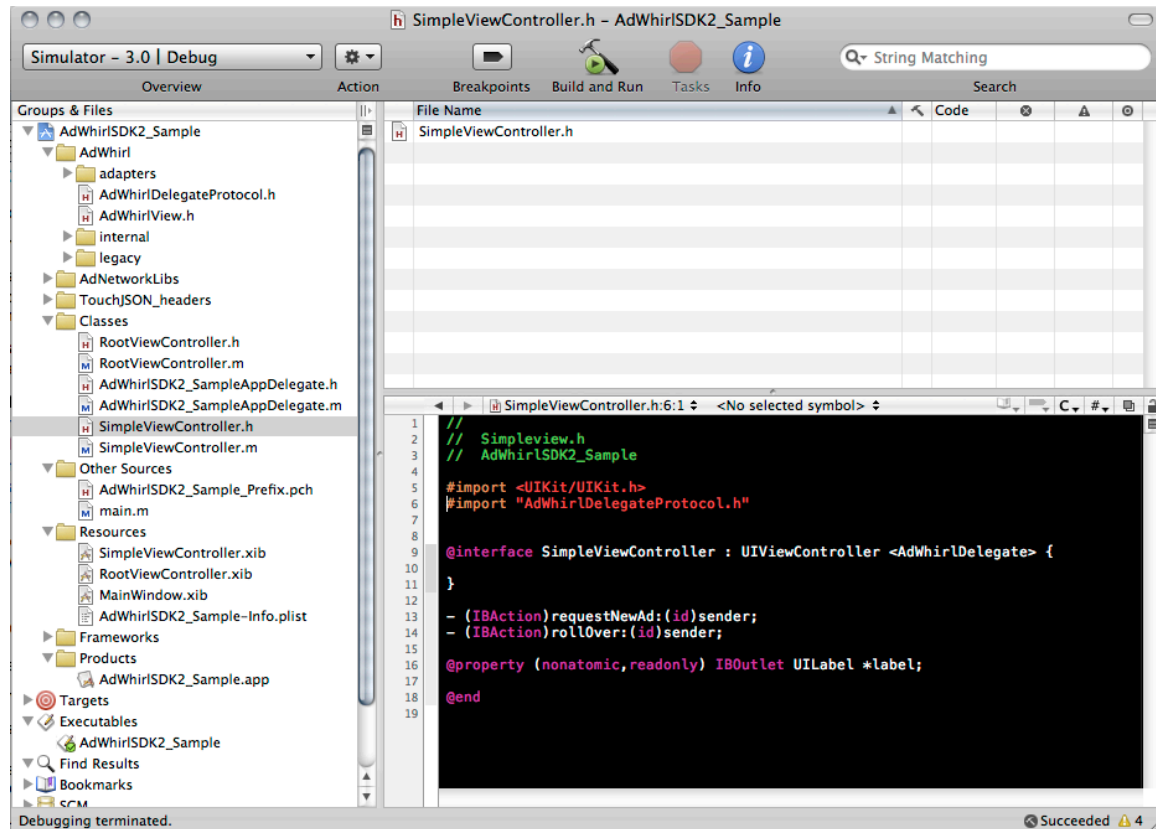
- iii. To avoid build errors, remove `AdFrameView.o` from under the ARM and x86 folders. Rename `libAdFrame.a` in each directory to include the architecture in the file name, for example, `libAdFrame-ARM.a` and `libAdFrame-X86.a`.



5. Remove the ad network adapters that you don't use. Under the `adapters/` group, you must remove the ad network adapters for ad networks that you did not integrate in step 4, or your project will not compile. In the extreme case, you can remove all files under `adapters/` except the `AdWhirlAdNetworkAdapter.h` file, in which case, your app can only run custom ads and generic notifications.

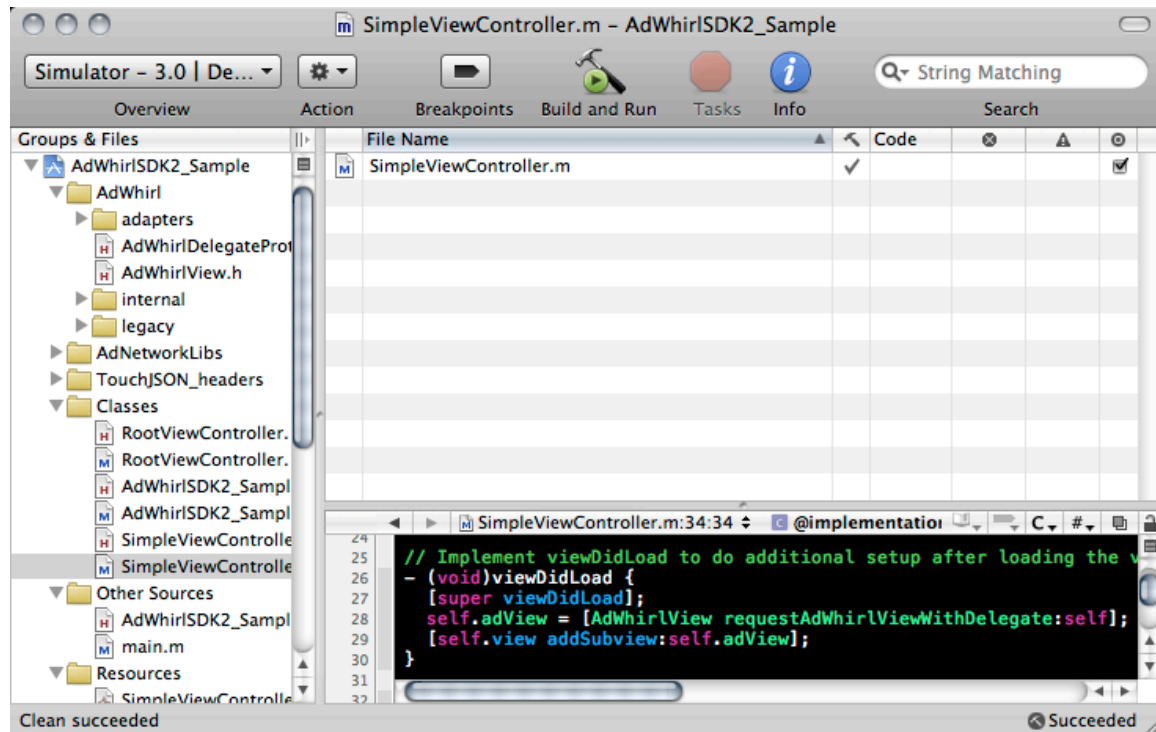


6. Implement the required `AdWhirlDelegate` methods in your code. If you have integrated with AdWhirl before and have implemented `AdRollerDelegate`, you do not have to change the code. This open source SDK is a drop-in replacement of ARRoller APIs and is backwards-compatible. At a minimum, you must implement the `adWhirlApplicationKey` method to return your AdWhirl application key. See the comments in `AdWhirlDelegateProtocol.h` for optional methods to implement.



7. Request an `AdWhirlView` in your code and add it to your view hierarchy. For example:

```
AdWhirlView *awView = [AdWhirlView  
requestAdWhirlViewWithDelegate:self];  
[self.view addSubview:awView];
```



8. To avoid build warnings (it is useful to have "Treat Warnings as Errors" turned on), you can uncheck static libraries from the target that are not the same architecture as those you are building.

