admob

AdMob iOS SDK

Installation Instructions

AdMob iPad / iPhone SDK Installation Instructions
The AdMob iPad / iPhone SDK contains the code necessary to install AdMob ads within your iPhone, iPod Touch, and iPad applications. The instructions below assume basic familiarity with Xcode.

Summary

AdMob ads are displayed in `UIViews`. Your implementation of the `AdMobDelegate` (AdMobDelegateProtocol.h) controls the features of the ad (e.g. color scheme) and receives notifications (e.g. the ad has loaded).

There are 4 sizes of ads:

- 320px X 270px
- 748px X 110px
- 488px X 80px
- 320px X 48px

We recommend that you use the 3 larger ad sizes when building their iPad applications because of iPad's higher screen resolution. These larger ad sizes will only be returned to iPads.

AdMob ads may be placed in multiple locations within your app: the main window of your app, in a table view, on menu screens, etc. We ask that you do not place ads near buttons or other areas where users tend to click. Such placements can lead to accidental ad clicks, which may result in AdMob temporarily disabling ad serving to such ad placements.

You may display the ad view when you begin requesting an ad or wait until the ad has loaded successfully (see the sample IB controller (AdViewController.m) for an example of how to do this).

Only request an ad if you intend to show it. Requesting ads and not showing them will hurt our ability to optimize your traffic and maximize your revenue.

Universal Binaries

If you are integrating the SDK into a universal binary, please make sure to use one AdMob publisher ID for your iPad application and a different AdMob publisher ID for your iPhone application.

We recommend you implement this by returning different publisher IDs in your code depending on the value returned from `UI_USER_INTERFACE_IDIOM()` .

Popovers (iPad Only)

If your iPad application uses popovers, please make sure to dismiss the popover whenever the AdMob SDK sends the `willPresentFullScreenModalFromAd:` callback.

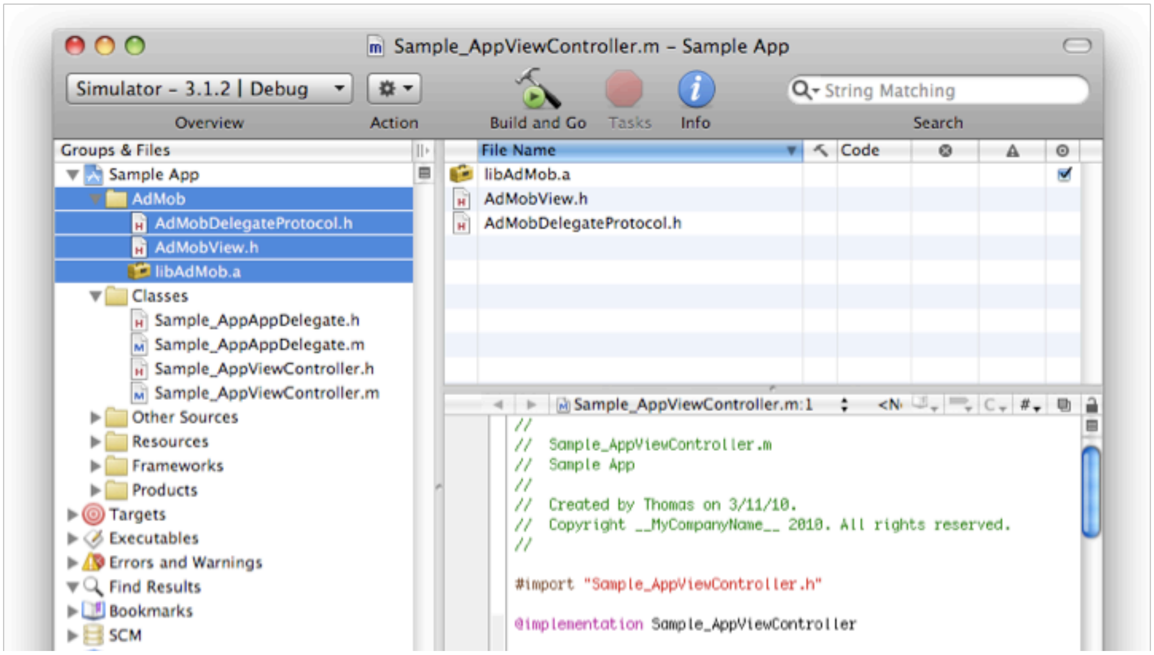
We provide an example of this implementation in `AdMobSampleAdsiPad` sample application (EarthquakeListViewController.m and DetailViewController.m). The sample code dismisses the popover whenever the AdMob SDK is about to present a modal view controller.

Instructions

Step 1

Add the AdMob library code and headers to your project. They are contained in the AdMob subdirectory. The files are:

- AdMobDelegateProtocol.h
- AdMobView.h
- libAdMob.a



```
//
//  Sample_AppViewController.m
//  Sample App
//
//  Created by Thomas on 3/11/10.
//  Copyright __MyCompanyName__ 2010. All rights reserved.
//

#import "Sample_AppViewController.h"

@implementation Sample_AppViewController
```

Figure 1: AdMob libraries added to project

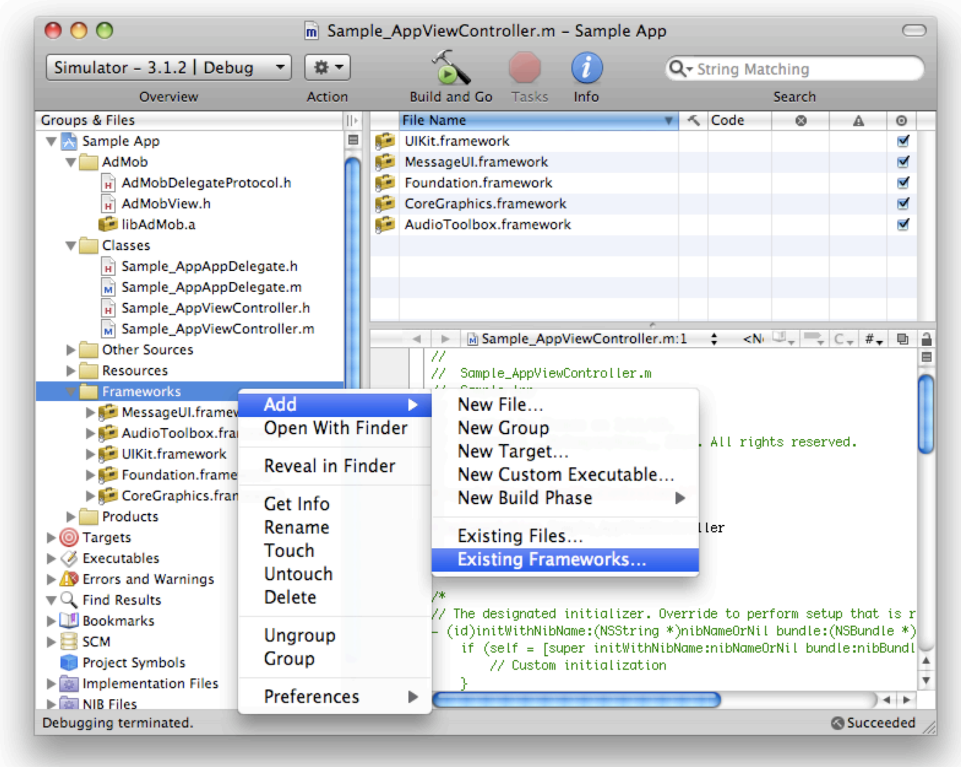
Note: If you are using Objective-C++, you may need a non-thumb build of `libAdMob.a`, which can be found in the extras directory, called `libAdMobNoThumb.a`

Step 2

Add the following frame works to your project:

- `AudioToolbox`
- `CoreGraphics`
- `MediaPlayer`
- `MessageUI`
- `QuartzCore`

These frameworks are required to compile.

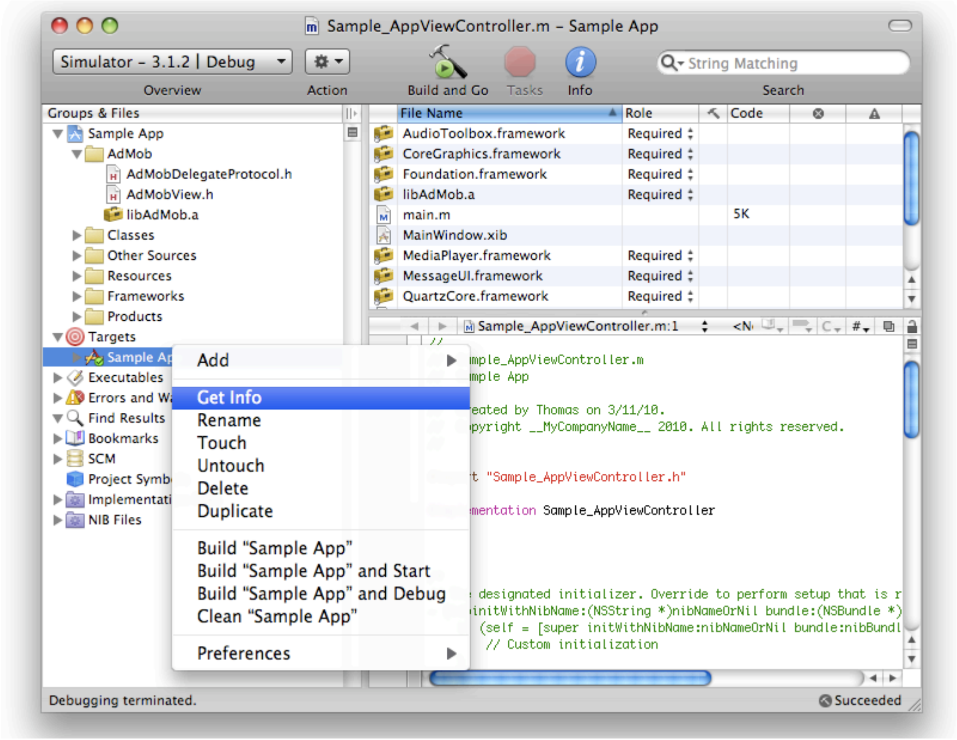


```
//
//  Sample_AppViewController.m
. All rights reserved.
ller
/*
// The designated initializer. Override to perform setup that is r
- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)
   if (self = [super initWithNibName:nibNameOrNil bundle:nibBundl
       // Custom initialization
   }
```

Figure 2: Frameworks > Add > Existing Frameworks

Note: If you would like to deploy to 2.x OS devices, you must weak link `MessageUI` or your application will not run on 2.x devices.

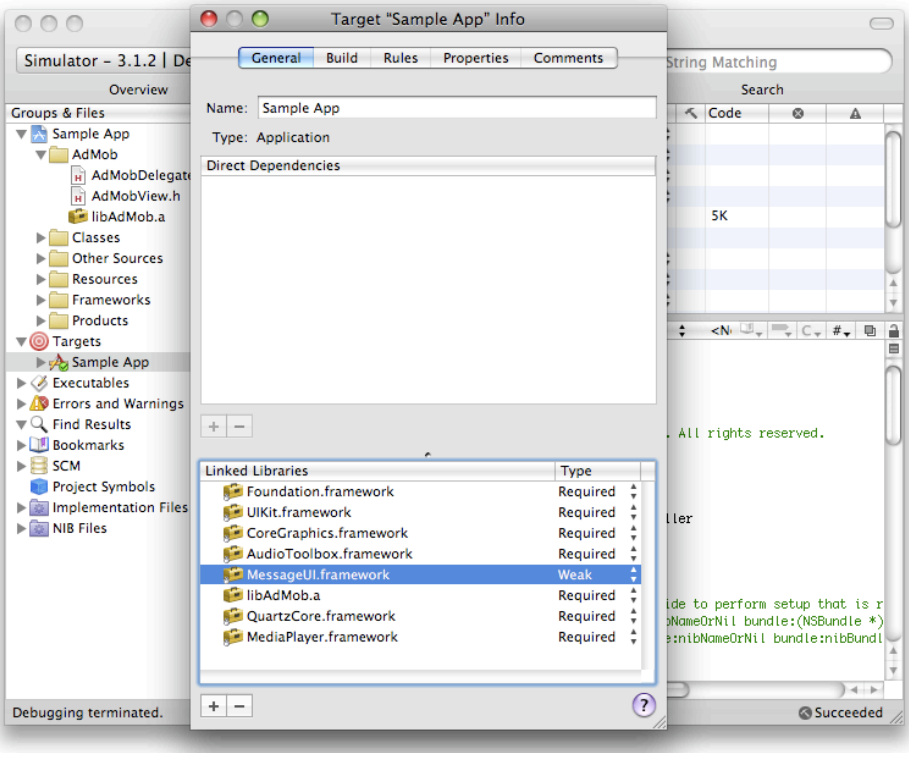
To weak-link `MessageUI`, expand the "Targets" group and right-click (Ctrl-click) on your target, and choose "Get Info". Under the “General” tab, change the Type for `MessageUI.framework` to "Weak."



File Name	Role	Code
AudioToolbox.framework	Required	
CoreGraphics.framework	Required	
Foundation.framework	Required	
libAdMob.a	Required	
main.m		5K
MainWindow.xib		
MediaPlayer.framework	Required	
MessageUI.framework	Required	
QuartzCore.framework	Required	

```
mple_AppViewController.m
mple App

eated by Thomas on 3/11/10.
pyright __MyCompanyName__ 2010. All rights reserved.

t "Sample_AppViewController.h"

mentation Sample_AppViewController

 designated initializer. Override to perform setup that is r
initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)
 (self = [super initWithNibName:nibNameOrNil bundle:nibBundl
  // Custom initialization
```

Figure 3: Expand "Targets" to access "Get Info" link



Name: Sample App

Type: Application

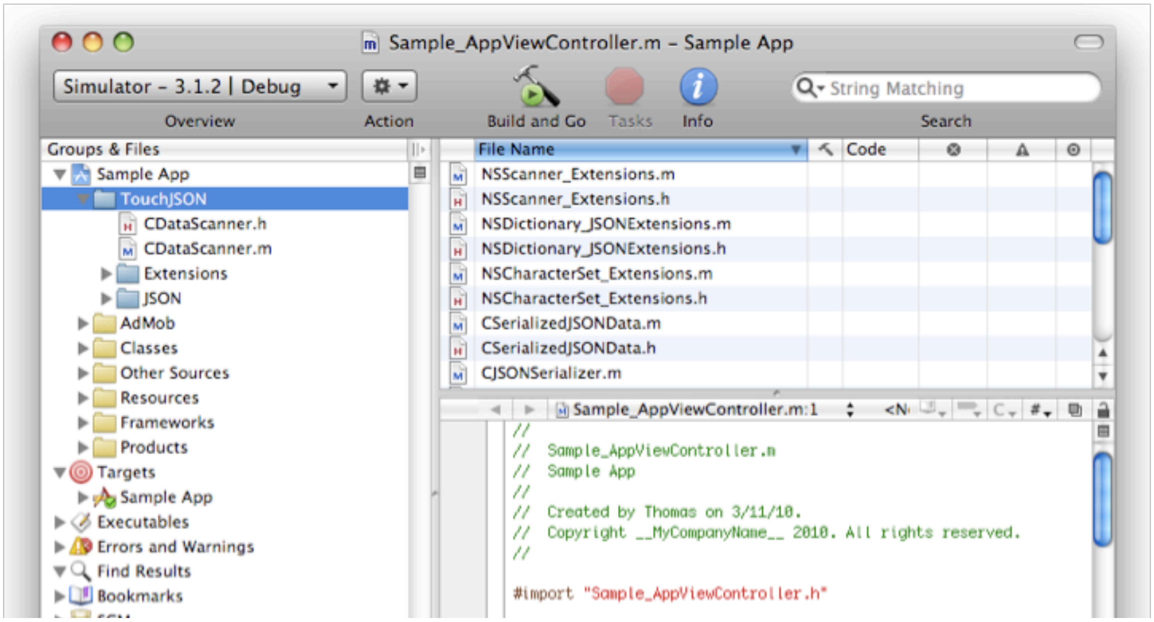
Linked Libraries	Type
Foundation.framework	Required
UIKit.framework	Required
CoreGraphics.framework	Required
AudioToolbox.framework	Required
MessageUI.framework	Weak
libAdMob.a	Required
QuartzCore.framework	Required
MediaPlayer.framework	Required

```
. All rights reserved.

ller

ide to perform setup that is r
NameOrNil bundle:(NSBundle *)
:nibNameOrNil bundle:nibBundl
```

Figure 4: Weak link Message UI Framework

Step 3

Add the `TouchJSON` source to your project. `TouchJSON` version 1.0.8 (April 23, 2010) is included with the AdMob SDK under the `TouchJSON` directory. Alternatively, you can include a compiled `TouchJSON` library sometimes provided with other third-party libraries.



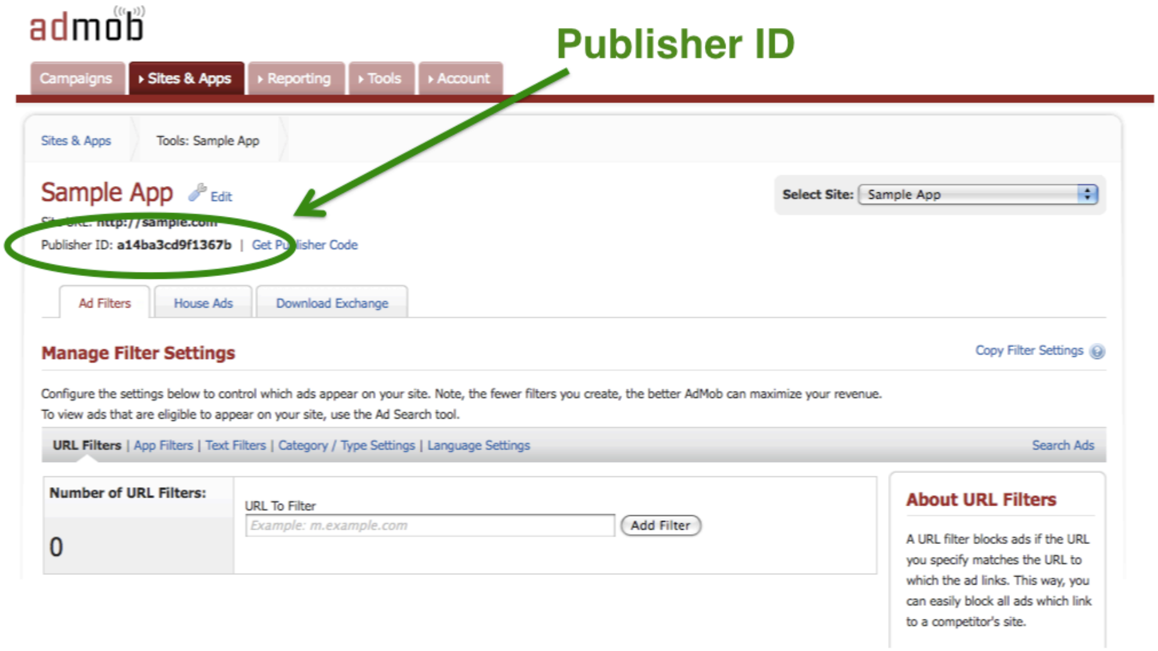
```
//
//  Sample_AppViewController.m
//  Sample App
//
//  Created by Thomas on 3/11/10.
//  Copyright __MyCompanyName__ 2010. All rights reserved.
//

#import "Sample_AppViewController.h"
```

Figure 5: Add TouchJSON folder to project

Step 4

Get your publisher ID from AdMob.com and set it in your delegate. To find your publisher ID, log into your AdMob account, select the “Sites & Apps” tab, and click on the “Manage Settings” link for your site. On this page, you can find your publisher ID (screenshot below).

If you are using the `IBSupport` files or running one of the sample projects, your publisher ID will be pre-filled for you.



Sample App

Site URL: http://sample.com

Publisher ID: a14ba3cd9f1367b | Get Publisher Code

Manage Filter Settings

Configure the settings below to control which ads appear on your site. Note, the fewer filters you create, the better AdMob can maximize your revenue. To view ads that are eligible to appear on your site, use the Ad Search tool.

Number of URL Filters: 0

URL To Filter: Example: m.example.com

About URL Filters

A URL filter blocks ads if the URL you specify matches the URL to which the ad links. This way, you can easily block all ads which link to a competitor's site.

Figure 6: Publisher ID on www.admob.com

Step 5 (iPad App)

Integrate AdMob ads in your application. We’ve included sample projects showing basic integration styles. The sample application can be found in the “/sample_projects/AdMobSampleAdsiPad” directory.

The three implementation styles are:

A) Add an ad to a view using Interface Builder

- Add `AdViewController.h` and `AdViewController.m` to your project (located in the IBSupport subdirectory).
- Open Interface Builder.
- Place a `UIView` where you want the ad to appear. Set the size of the view to the size of the ad you want to appear (e.g. 320X270, 748X90, 488X80 or 320X48). Our sample sets the `UIView` to 320X270 to request the 320X270 ad unit.
- Add an Object, and change its type to `AdViewController`.
- Set the view outlet of the `AdViewController` to your `UIView`.
- Set the `currentViewController` outlet of the `AdViewController` to the `UIViewController` owning the `xib`.
- In the UIViewController that owns the xib, create an AdViewController IBOutlet that retains. Establish a connection from the owner UIViewController to the AdViewController. Remember to release the AdViewController in the UIViewController's dealloc method.
- Edit `AdViewController.m` to make sure that your publisher ID and other options are set correctly.

B) Add an ad to a view programmatically

- Create an AdMob delegate class; it can be as simple as a class that only implements `-publisherIdForAd:` and `-currentViewControllerForAd:`.

- Add the following code to your view creation (modify as appropriate):

```
adView = [AdMobView requestAdOfSize:ADMOB_SIZE_748x110
withDelegate::<your delegate>]]; // start a new ad
request

adView.center =
CGPointMake(self.view.bounds.size.width/2.0,
self.view.bounds.size.height-
(adView.bounds.size.height/2));  // position ad at the
bottom of the view and centered horizontally

[self.view addSubview:adView];// attach the ad to the
view hierarchy; self.window is responsible for
retaining the ad
```

C) Add an ad to a tableView

- Create an AdMob delegate class; it can be as simple as a class that only implements `-publisherIdForAd:` and `-currentViewControllerForAd:`.
- Add to your `-tableView:cellForRowAtIndexPath:` method (modify as appropriate):

```
[cell.contentView addSubview:[AdMobView
requestAdWithDelegate:<your delegate>]];
```

Step 5 (iPhone)

Integrate AdMob ads in your application. We've included sample projects showing basic integration styles. Please read both the README file and the sample projects contained with our SDK for more information on how to integrate AdMob ads into your application.

The three implementation styles are:

A) Add an ad to a view using Interface Builder

- Add `AdViewController.h` and `AdViewController.m` to your project (located in the `IBSupport` subdirectory).
- Open Interface Builder.
- Place a 320x48 `UIView` where you want the ad to appear.
- Add an Object, and change its type to `AdViewController`.
- Set the view outlet of the `AdViewController` to your `UIView`.
- Set the `currentViewController` outlet of the `AdViewController` to the `UIViewController` owning the `xib`.
- Edit `AdViewController.m` to make sure that your publisher ID and other options are set correctly.

B) Add an ad to a view programmatically

- Create an AdMob delegate class; it can be as simple as a class that only implements `-publisherIdForAd:` and `-currentViewControllerForAd:`.
- Add the following code to your view creation (modify as appropriate):

```
AdMobView *ad = [AdMobView requestAdWithDelegate:<your
delegate>]; // start a new ad request

ad.frame = CGRectMake(0, 432, 320, 48); // set the
frame, in this case at the bottom of the screen

[self.window addSubview:ad]; // attach the ad to the
view hierarchy; self.window is responsible for
retaining the ad
```

C) Add an ad to a tableView

- Create an AdMob delegate class; it can be as simple as a class that only implements `-publisherIdForAd:` and `-currentViewControllerForAd:`.
- Add to your `-tableView:cellForRowAtIndexPath:` method (modify as appropriate):

```
[cell.contentView addSubview:[AdMobView
requestAdWithDelegate:<your delegate>]];
```

For More Information

You can connect with other developers using the AdMob iOS SDK here:

http://groups.google.com/group/admob-iphone-sdk

If you have any other questions, please feel free to contact support@admob.com.