Ripple Mobile SDK

Smart Media for Smarter Screens

Your How-to Guide

Using Ripple SDK Ads within iOS Apps

Version: ripple1.0.0

Table of Contents

1. Introduction

1.1. Disclaimer

This document is provided “as-is.” Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Ripple product. You may copy and use this document for your internal, reference purposes.

1.2. App Market places

The Apple App Store apply conditions on app developers that they must comply with, when participating or uploading apps to these marketplaces. Please note, the inclusion of this SDK does not transfer that responsibility and it remains the responsibility of the App developer to achieve compliance with any app Apple App Store conditions.

Where possible, our SDK include features and functions to support an App developers efforts with the Apple App Store compliance and we strongly recommend including or utilizing those features where possible.

1.3. Before you start

Register as Ripple App publisher

Please send all details below to developer@ripple.ad to sign up to Ripple Ad Network:

- First Name
- Last Name
- Company Name
- Address
- Phone Number
- Your Email
- Country
- Your Domain Name
- Where will you be running our ads? (Enter mobile site or app address)

Your account will be processed within 1 working day so please bear with us. If approved, we will send your publisher log in details within 1 working day.

Each Ad has a unique alias, network Id, subNetwork Id that you should have available for inclusion the following code.

Get the Ripple Ads SDK

The Mobile SDK provides you with all of the necessary tools for integrating in-app advertising for iOS. You can get the Ripple SDK from the Ripple Publisher portal for App publisher including this guide and a sample app which can be used for showcase purposes and for testing along with the program library. You will need to add these files into your App project directory as indicated in the next section. This SDK includes the header files and the framework (rippleAd.Framework). You will need to add these files into your app project directory as indicated in the next section.

Incorporating the SDK

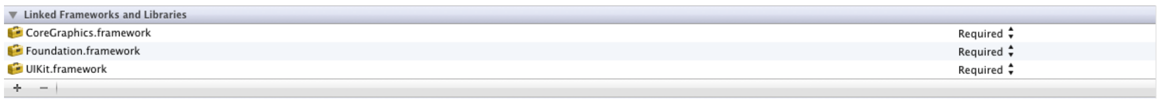
It is recommended that you use XCode 4.5 and above.

2. Quick start: Ad Integration Guide

To add banner ad in your application, please follow steps below:

Step 1: Configure the required frameworks on XCode

In the XCode IDE (v4.5 and above), click on your project and then click on the target. This will bring up a list where you can add static libraries and frameworks to the selected target.



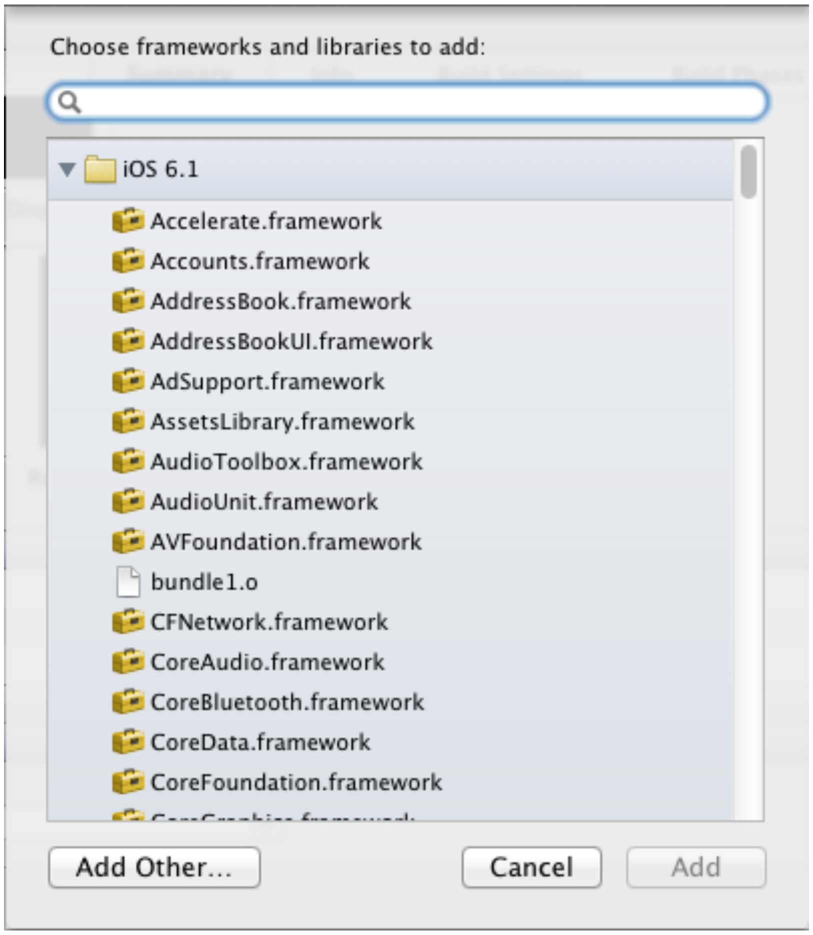
Linked Frameworks and Libraries	
CoreGraphics.framework	Required
Foundation.framework	Required
UIKit.framework	Required

In this window click on the "+" sign under the "Linked Frameworks and Libraries" drop-down and you should be able to select following frameworks (multi-select by holding down Apple key while selecting):

- AdSupport.framework (Only for iOS 6.x onward)
- CoreGraphics.framework
- CoreLocation.framework
- CoreMotion.framework
- CoreTelephony.framework
- EventKit.framework
- EventKitUI.framework
- Foundation.framework
- MediaPlayer.framework
- MessageUI.framework
- SystemConfiguration.framework
-
- UIKit.framework
- libxml2.dylib
- libz.dylib

Step 2: Add the Ripple static library

To add the Ripple Ad framework, follow the same steps as above and then click “Add Other…” and select the “rippleAd.Framework” file from your project folder.



Step 3: Add the Ripple static library resource

To add the Ripple Ad Framework resource file, simply drag rippleAd.bundle file into your XCode project.

Step 4: Test your ads using the Test configuration

The simplest way to run a Ripple ad unit in your App is to add the code in your App's view controller as seen by the example code below:

```
@property (strong, nonatomic) IBOutlet BannerView *bannerView;
```

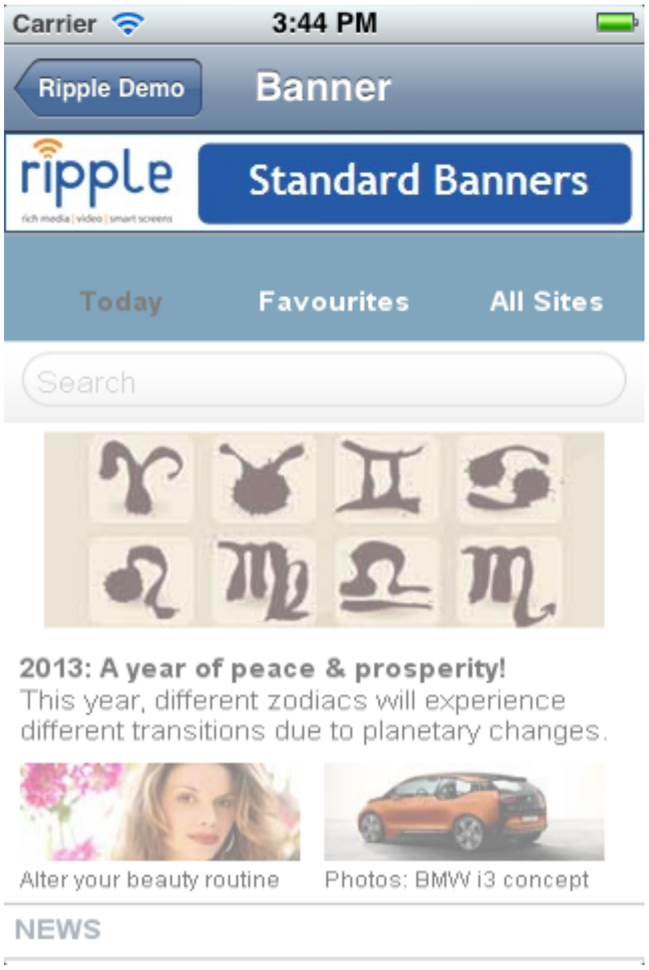
```
- (void)viewDidLoad
{
    [super viewDidLoad];

   // Test configuration please change below with your own configuration
   // after testing.
    self.bannerView.alias=@"SDK_Banners-Default-1";
    self.bannerView.networkID=1333;
    self.bannerView.subnetworkID=1;
   // End test configuration

   self.bannerView.refreshInterval=60;
   [self.bannerView loadAd];
}
```

Step 5: Testing Banner Ad

Run your application and you should see something like the diagram below:



2013: A year of peace & prosperity!
This year, different zodiacs will experience different transitions due to planetary changes.

Alter your beauty routine

Photos: BMW i3 concept

NEWS

After you have tested your application please change this test configuration to your own publisher configuration as stated in "Step 4: Test your ads using the Test configuration". Then please re-run your application, to make sure that you can pull out live ads. Below is the parameters that must be changed:

Standard Banner Test Ad

Network Id: 1333

Sub Network Id: 1

Alias: RippleSDK_SB-bottom-5

3. Advance Banner Ad Integration Guide

3.1. Pause and Resuming Ads

Pause and Resume Ad functionality is available through the SDK as well. To pause the ad section simply call pauseAd function on the BannerView instance in your App. Pausing an Ad means no ad request will be sent to the Ad servers even though the loadAd has been called. To resume the ad section call resumeAd function on the BannerView instance.

3.2. Refreshing Ads

Refreshing functionality is available by calling loadAd.

3.3. Implementing Event Listeners

For Ads, it is possible to listen for specific events with the Ad listener interface and write app specific code at these events. Event listeners exist for:

- Ad Load – When the ads are loaded on for the specific placement.
- NoAds – No Ads were returned from the server.
- Ad Switch – When the page is shown for the number of seconds configured in the refreshInterval for the BannerView instance, a new ad will be requested. When this happens this trigger will be called.

- Ad Pause – This will be called after you have called the pauseAd() function mentioned above.
- Ad Resume – This will be called after you have called the resumeAd() function mentioned above.
- Ad Fail – This will be called when there is an error with the system, which may include loss of internet connection, and so forth.

Firstly your view controller class needs to implement the AdDelegate class, as shown in the example below:

```
#import <ripplead/AdDelegate.h>
@interface ViewController : UIViewController <AdDelegate>
```

Once your class implements AdDelegate, All 6 abstract methods will need to be implemented in your class, as shown below. To utilize these methods just add the app specific code you want to implement.

```
- (void) onAdsLoad {
        NSLog(@"on ads load");
}
- (void) onNoAds {
        NSLog(@"No ads available");
}
- (void) onAdsSwitch {
        NSLog(@"On ads switch");
}
- (void) onAdsPause {
        NSLog(@"on ads pause");
}
- (void) onAdsResume {
        NSLog(@"on ads resume");
}
- (void) onAdsFail {
        NSLog(@"on ads fail");
}
```

Now, finally when initializing the BannerView class, you will need to pass the AdDelegate to it, as shown in the example below:

```
self.bannerView.delegate = self;
```

4. Interstitial Ad Integration Guide

To add interstitial ad in your application, firstly you need to follow the steps 1 to 3 on section 4. Once steps 1 to 3 on section 4 is implemented, please follow the steps below:

Step 1: Interstitial Ad Integration

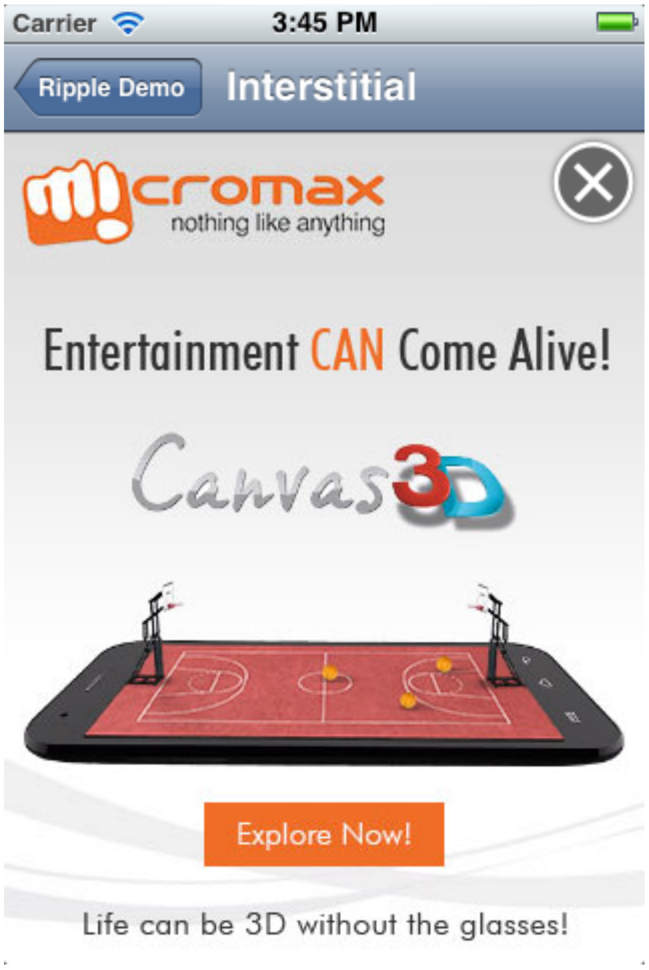
The simplest way to run a Ripple ad unit in your App is to add the code in your App's view controller as seen by the example code below:

```
@property (strong, nonatomic) IBOutlet InterstitialView *interstitialView;
```

```
- (void)viewDidLoad
{
    [super viewDidLoad];
   // Test configuration please change below with your own configuration
   // after testing.
   self.interstitialView.alias=@"RSDK_Inter-bottom-5";
    self.interstitialView.networkID=1333;
    self.interstitialView.subnetworkID=1;
   // End test configuration
   [self.interstitialView loadAd];
}
```

Step 2: Testing Interstitial Ad

Run your application and you should see something like the diagram below:



After you have tested your application please change this test configuration to your own publisher configuration as stated in "Step 1: Interstitial Ad Integration". Then please re-run your application; to make sure that you can pull out live ads. Below is the parameters that must be changed:

Interstitial Test Ad

Network Id: 1333

Sub Network Id: 1

Alias: RSDK_Inter-bottom-5

5. Ripple Ads for Mobile FAQ

General Information

1. Q: What ads formats does the SDK support?

   A: This SDK compliant with IAB's Rich Media Standard MRAID 1.0 and MRAID 2.0.

Development

1. Q: Where can I find more help content specific to the Ripple network?

   A: You can contact our support team at support@ripple.ad to get more help

2. Q: I think I've implemented everything correctly, so why am I not seeing ads?

   A: While maintaining the highest possible fill rate is one of our top priorities, we may not always have an ad available for every ad request. This can be especially common during development, when ad requests are typically made infrequently from a small number of users and devices. When apps are newly registered on Ripple it may also take some time and several requests before impressions are consistently delivered. Developers generally see more consistent results once they have released their app and ad requests arrive more frequently from a more diverse user base.