Taptica iOS Mraid V2 SDK

Product Release 1.2

Overview

Taptica is the leading mobile user acquisition platform delivering high performance solutions to advertisers and publishers, reaching over 200M users worldwide. We work with 150 + advertisers including brands and app developers.
The Taptica iOS SDK provides a simple way to integrate ads and monetize your iOS applications. With the SDK, you can easily integrate rich media (MRAID 2.0) and video ads into your application.

System requirements

The Taptica iOS SDK requires iOS 5.1 or above, Apple LLVM compiler and the following list of frameworks to run:

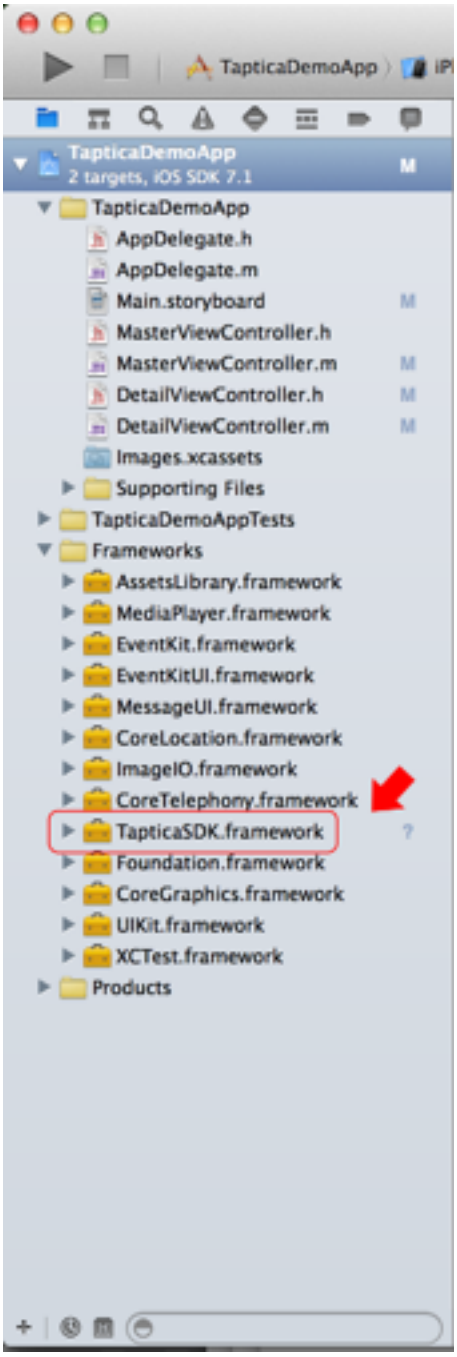
- AssetsLibrary.framework
- Foundation.framework
- UIKit.framework
- MediaPlayer.framework
- EventKit.framework
- EventKitUI.framework
- MessageUI.framework
- CoreLocation.framework
- CoreGraphics.framework
- ImageIO.framework
- CoreTelephony.framework

SDK contents

- Framework
- Sample – sample usage/test app
- Documentation

Integration guide

1. Drag the TapticaSDK.framework to the Frameworks Folder (See the image blow)



2. Verify that TapticaSDK.framework has been added to the Link Binary with Libraries Build Phase for the targets you want to use the SDK with

    a. Select your Project in the Project Navigator

    b. Select the target you want to enable the SDK for

    c. Select the Build Phases tab

    d. Open the Link Binary with Libraries Phase

    e. If TapticaSDK.framework is not listed, drag and drop the library from your Project Navigator to the Link Binary with Libraries area

    Repeat Steps b - e until all targets you want to use the SDK with have the SDK linked

3. Add required frameworks to your link binary with libraries build phase

    a. Select your Project in the Project Navigator

    b. Select the target you want to enable the SDK for

    c. Select the Build Phases tab

    d. Open the Link Binary with libraries phase

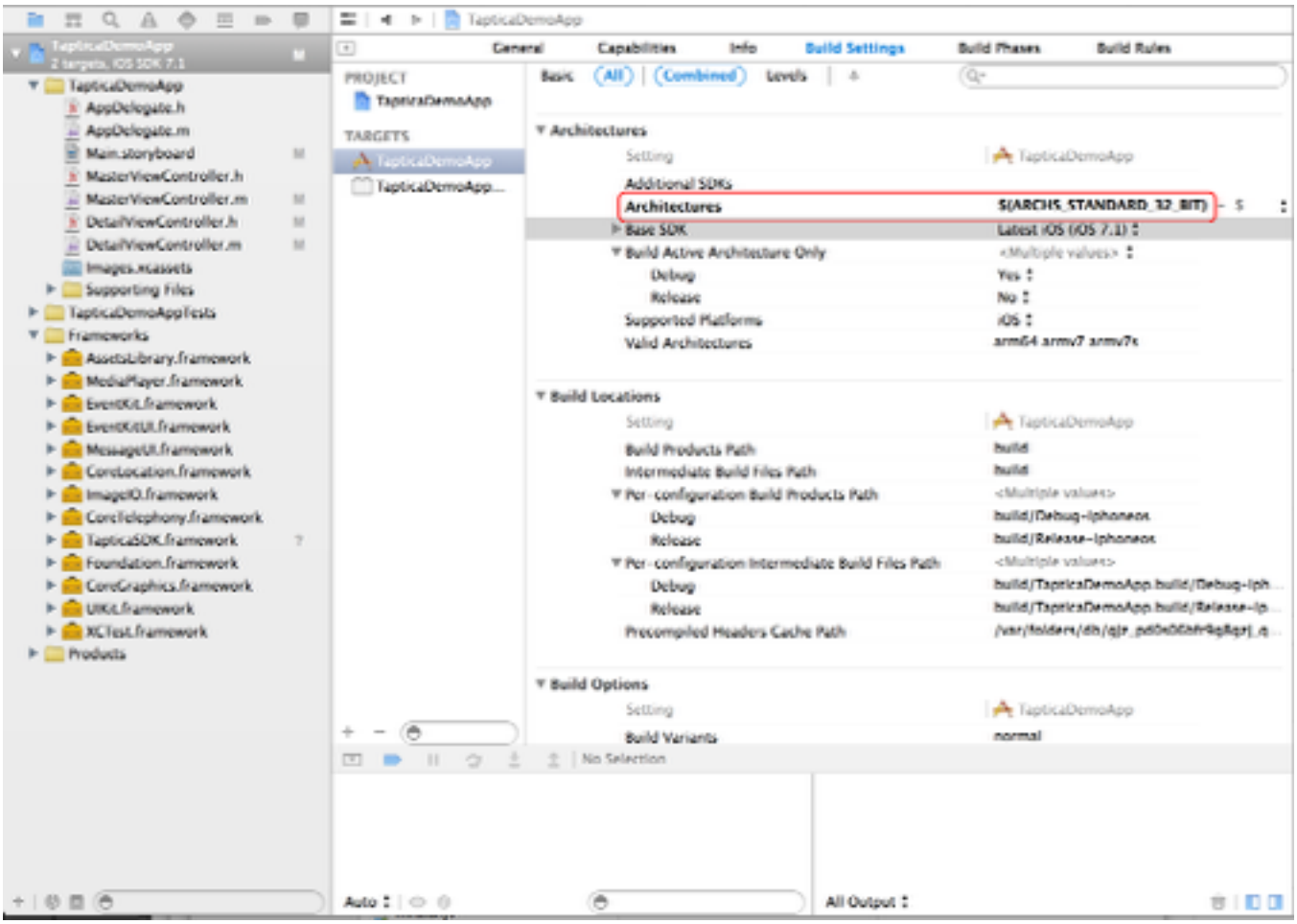
    e. Click the + to add a new framework

    f. Find framework in the list and add it

    Repeat Steps a - f for all targets.

4. Replace the content of Architectures in Build Settings with:

$(ARCHS_STANDARD_32_BIT)



Using the SDK

Those samples of code shows you how to add several types of advertising:

Banners

```
// Set your own frame for AdView
        CGRect rect = self.view.frame;
        rect.size.height = 60;


        TAdView * adView = [[TAdView alloc] initWithFrame:rect withDelegate:nil];// Set frame and
delegate
        adView.useInternalBrowser = YES; // Set it if you want to open ads links in the embedded
browser
        adView.backgroundColor = [UIColor clearColor];
        adView.autoresizingMask = UIViewAutoresizingFlexibleWidth;
        adView.adType = 2; // Ad type for banner
        adView.applicationId = @"8fbd13dd-0d52-4cf1-bd01-45017dba0e73"; // Your application ID
        [self.view addSubview:adView];
        [adView update]; //Ad will start updates
```

Interstitial

```
TAdView * adView = [[TAdView alloc] initInterstitial:self]; //Alloc and init Interstitial with self delegate
adView.useInternalBrowser = YES; // Set it if you want to open ads links in the embedded browser
adView.adType = 3; // Ad type for Interstitial
adView.applicationId = @"2b04d2d6-bcb7-4f01-8fee-8360b0f05275"; // Your application ID
[adView update]; //Ad will start updates
```

You have to use delegate methods to show or hide interstitial

Delegate for Interstitial

```
// Sent after an ad was downloaded and rendered

- (void) didReceive:(TAdView *)adView {
...  // Your code
[adView showInterstitial]; // Call to show Interstitial
... // Your code
}
//Sent when close button is pressed by the user
- (void)closeButtonPressed:(TAdView *)adView {
... // Your code
[adView closeInterstitial]; // Close Interstitial
... // Your code
}
```

Taptica Video Ads

```
TAdView * adView = [[TAdView alloc] initWithFrame: CGRectZero withDelegate:nil];
// Note that, you can set any frame, no matter, video ad will be opened as modal view of your root view controller. Video would occupy all screen.

adView.useInternalBrowser = YES;
// Set it if you want to open video ads links in the embedded browser

adView.adType = 5; // Ad type for video
adView.applicationId = @"d1b1de27-50f5-40a6-ace3-e35dcf2b65ad "; // Your application ID
[self.view addSubview:adView];
[adView update]; //Ad will start updates
```

Don’t forget to reset the ad when it’s no longer need, or when you want to show a different type of advertising.

```
[adView reset]; // Ad will stop updates
[adView removeFromSuperview];
```

Reset method doesn’t reset the delegate - you will need to do it yourself:

```
[adView setDelegate:nil];
```

Additional information about classes and delegate methods can be found in the API documentation.

Advertiser Events

Publishers can increase their CPI payout with advertiser Events. This function allows you to get a higher payout for ad clicks performed in your app by serving relevant ads to your paying users.
The SDK enables you to share data with Taptica, on users who have made purchases and have spent money in your app (your most profitable users). We will be able to efficiently target those users with relevant ads that they are highly likely to click on and in return you will get a higher CPI payout. This helps publishers create a stronger avenue for monetization and create a better user experience.

Any data shared with Taptica will be used for the purpose of serving ads exclusively to your app and will not be used elsewhere!

Tracking events

You should call register method of Service in advance because initializing process takes some time. You need to provide Service with advertiser ID and Itunes ID params, also you can set additional LogLevel param for logging and delegate for receiving notifications about sending event process.

```
[[TTrackerService sharedInstance] registerTrackerService:advertiserID ItunesID:itunesID];
```

```
[[TTrackerService sharedInstance] setLogLevel:TTrackerServiceLogLevel_Mid
AndDelegate:self];
```

When you are ready to send event you can initialize TAdditionalParameters object (Gender and age).

```
TAdditionalParameters * params = [[TAdditionalParameters alloc] initWithAge:age Gender:gender];
```

You can send Purchase event. (requires NSNumber purchase amount)

```
[[TTrackerService sharedInstance] sendPurchaseEvent:params withAmount:amount];
```

Or Action event. (requires TTrackerActionType type)

```
[[TTrackerService sharedInstance] sendActionEvent:params withType:type];
```

For a more complete example of usage please see our sample usage/test app. All the necessary information about classes can be find in the API documentation.