iOS Google Mobile Ads Mediation Adapter

Overview

The AdColony iOS Google Mobile Ads Mediation Adapter can be used by developers to incorporate AdColony video ads into the Google Mobile Ads mediation platform. The following instructions assume you have already integrated the Google Mobile Ads SDK and are ready to add the AdColony adapter and SDK to your Xcode project. If you have not already integrated the Google Mobile Ads SDK into your application, please refer to the documentation.

Integration Instructions

To add AdColony to your mediation layer, you will need to create an app on the AdColony dashboard. For help with setting up an AdColony app, follow steps 1-3 of Setting Up AdColony Apps.

Once you have created your AdColony app, you can add the AdColony network to your mediation layer. Note that you will need to enter your AdColony app and zone IDs into the AdMob dashboard when you add the AdColony network.

Download the AdColony AdMob Mediation Adapter as well as the AdColony iOS SDK.

Copy the **AdColony.framework**, **libAdapterSDKAdColony.a**, **GADMAdapterAdColonyInitializer.h**, and **GADMAdapterAdColonyExtras.h** files into your Xcode project and add them to all targets that will utilize AdColony.

Open **Target > Build Settings > Architectures** and confirm that your Xcode project uses Base SDK version 6.0 or greater and that it does not use the armv6 architecture.

Ensure the following libraries and frameworks have been added to **Target > Build Phases > Link Binary With Libraries**:

- libz.1.2.5.dylib
- AdColony.framework
- AdSupport.framework (Set to Optional)
- AudioToolbox.framework
- AVFoundation.framework
- CoreGraphics.framework
- CoreMedia.framework
- CoreTelephony.framework
- EventKit.framework
- EventKitUI.framework
- MediaPlayer.framework
- MessageUI.framework
- QuartzCore.framework
- Social.framework (Set to Optional)
- StoreKit.framework (Set to Optional)
- SystemConfiguration.framework
- WebKit.framework (Set to Optional)

Ensure the following flags have been added to **Target > Build Settings > Other Linker Flags**:

- ObjC
- fobjc-arc (this allows AdColony to use ARC even if your project does not)

Optimizations

To ensure AdColony video ads are available as often as possible, you can initialize the AdColony SDK early on in the application lifecycle and outside of the Google Mobile Ads SDK. To do so, import **GADMAdapterAdColonyInitializer.h** into your AppDelegate and insert the following code into the `application:didFinishLaunchingWithOptions:` method:

```
[GADMAdapterAdColonyInitializer startWithAppID:@"YOUR_ADCOLONY_APP_ID"
                                       andZones:@[@"Zone_ID_1", @"Zone_ID_2"]
                                    andCustomID:@”YOUR_CUSTOM_ID”];
```

Using Multiple Zones

The AdColony Adapter supports the use of multiple zones with your placement. A few extra steps are necessary, however.

When you add AdColony to your ad placement in the AdMob dashboard, you will need to enter a semicolon-delimited list of your AdColony zone IDs (see example below). Note that the first member of this list will be used as the default zone from which ads are played.

```
vzf7a1a2d123fd489bab37b0;vz23953bee9a8c4a2eb9d2dd;vz3d55f55e042f49259094ad
```

To play video ads from different AdColony zones, you will need to import **GADMAdapterAdColonyExtras.h** and create a **GADMAdapterAdColonyExtras** object (see example below).

```
_extras = [[GADMAdapterAdColonyExtras alloc] initWithZone:@"YOUR_ZONE_ID"];
```

The AdColony Adapter will use the ‘currentZone’ property of the extras object to determine which zone to play ads from. Note that you will have to register your extras object with each ad request you make (see example below).

```
_interstitial = [[GADInterstitial alloc] init];
_interstitial.delegate = self;
_interstitial.adUnitID = @"YOUR_AD_UNIT_ID";
_extras.currentZone = @"YOUR_ZONE_ID"
_request = [GADRequest request];
[_request registerAdNetworkExtras:_extras];
[_interstitial loadRequest:_request];
```

Displaying Instant-Feed Ads

The AdColony Adapter supports the mediation of AdColony Instant-Feed ads through the Google Banner API. The AdColony Adapter supports the Medium Rectangle banner size as well as custom ad sizes that meet the following constraints:

- Minimum width: 300px
- Minimum height: 200px
- Aspect ratios between 4:3 and 16:9

Requesting a banner size that does not meet these requirements will never give an ad.

To request an Instant-Feed ad using the Google Banner API, make a banner request like follows:

```
GADMAdapterAdColonyExtras* extras = [[GADMAdapterAdColonyExtras alloc]
                                       initWithZone:kAdColonyAdZone1];
_bannerView = [[GADBannerView alloc] initWithAdSize:kGADAdSizeMediumRectangle];
_bannerView.delegate = self;
_bannerView.adUnitID = kDFPAdUnitID;
_bannerView.rootViewController = self;
_bannerRequest = [GADRequest request];
[_bannerRequest registerAdNetworkExtras:extras];
[_bannerView loadRequest:_bannerRequest];
```

To properly configure and display the AdColony Instant-Feed ad, you must implement the `adViewDidReceiveAd`: delegate method:

```
- (void)adViewDidReceiveAd:(GADBannerView *)view {
      NSString* adNetworkName = view.adNetworkClassName;

      if ([adNetworkName isEqualToString:@"GADMAdapterAdColony"]) {

            // Cast the returned view to an AdColonyNativeAdView
            AdColonyNativeAdView* nativeAdView =
                (AdColonyNativeAdView*)[view mediatedAdView];

            // Configure AdColonyNativeAd view and add it to your view
            ...
      } else {
            // Display banner as normal
            ...
      }
}
```

See the AdColony Instant-Feed API to for more information on how to configure and display the advertisement.

Videos-for-Virtual-Currency

The AdColony Adapter supports the AdColony V4VC system through the Google RewardBasedVideo API. To request a V4VC ad use the following:

```
NSString *userID = @”YOUR_USER_ID”; //a unique string to identify this user
GADRequest *request = [GADRequest request];
GADMAdapterAdColonyExtras* extras =
       [[GADMAdapterAdColonyExtras alloc] initWithZone:kAdColonyAdZone1];
[[GADRewardBasedVideoAd sharedInstance] loadRequest:request
       withAdUnitID:kDFPAdUnitID userID:userID];
```

The `setCustomID:` method of AdColony is often used with the AdColony V4VC system in order to bind a unique user identifier to a specific reward. The adapter will call this method with the value passed to the `userID` parameter of the `loadRequest:withAdUnitID:userID:` method of `GADRewardBasedVideoAd`. If you are using the **GADMAdapterAdColonyInitializer** as described in the **Optimizations** section above, you must provide the custom ID you wish to use in order to ensure that the correct custom ID is used. The adapter will still call `setCustomID:` with the value provided as `userID`, but it is not guaranteed to be updated immediately.

Notes

- The AdColony SDK only supports iOS 6 and higher and will disable itself on any previous versions.