Flurry Advertising

Flurry iOS Adapter for MoPub

Adapter version 6.3.0.r1
Updated: 04/03/2015

Mediate Flurry through MoPub

To integrate Flurry as the Custom Native Network in MoPub ad serving flow, you need Custom Event Class code incorporated into your application in addition to Flurry SDK. Please follow the three quick steps:

1. Integrate Flurry SDK and Flurry adapter for MoPub code into your app
2. Configure Flurry's Ad space(s)
3. Configure MoPub to mediate Flurry

1. Integrate Flurry SDK and custom code into your app

1. If your application is not tracked by Flurry, create a new application on Flurry's dev portal. After logging into https://dev.flurry.com, select Applications tab and from top right-hand corner select Add New Application. In case your application is already tracked by Flurry, you can download the latest SDK from the adjacent top right-hand link.

2. Download the Flurry iOS SDK. Keep a reference to the API Key found on the download page. This will identify your app in the Flurry system.

3. In the finder, drag Flurry/ and FlurryAds/ into your project's file folder. If you are maintaining the structure advised by MoPub for 3rd party ad networks (https://github.com/mopub/mopub-ios-sdk/wiki/Getting-Started#adding-third-party-ad-networks) , add Flurry folder under AdNetworkSupport. Within Flurry folder add an additional one for SDK and drag Flurry/ and FlurryAds/ into it.

4. Add Flurry custom event files (.h and .m):

- FlurryBannerCustomEvent
- FlurryInterstitialCustomEvent

- FlurryNativeCustomEvent
- FlurryNativeAdAdapter
- FlurryMPConfig

5. Initialize Flurry by changing the value of "FlurryAPIKey" defined at the top of the file FlurryMPConfig.h. The API key is available in the Flurry SDK package and from the dev portal.
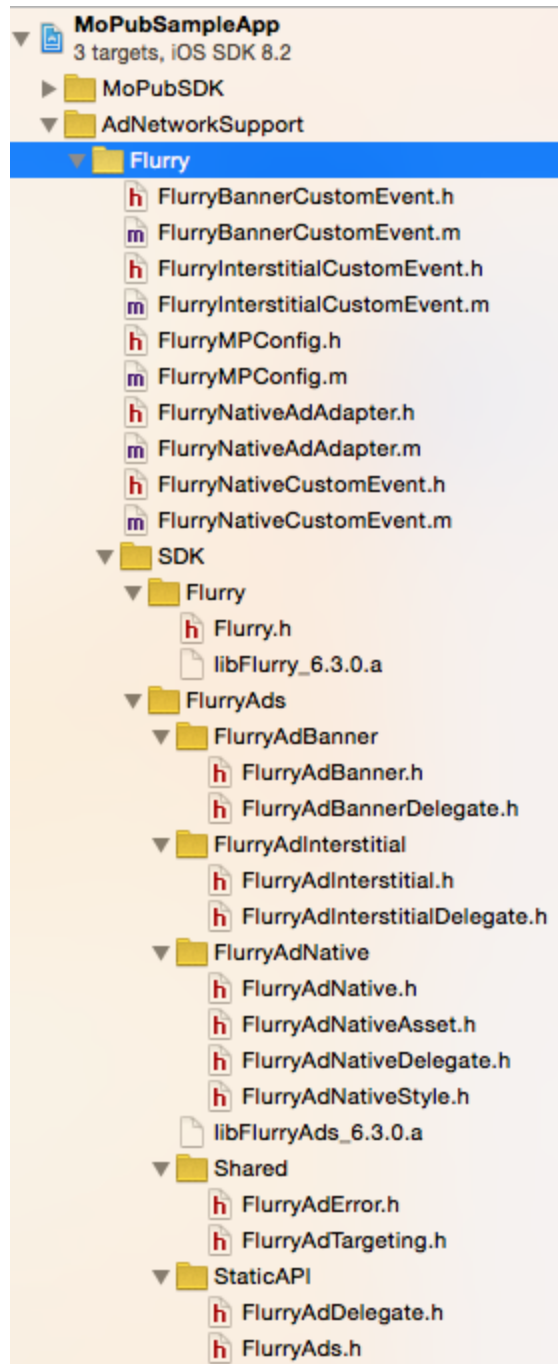
6. The frameworks that Flurry requires are:
    - AdSupport.framework (Mark as Optional)
    - CoreGraphics.framework
    - Foundation.framework
    - iAd.framework (required if you would like to integrate iAD serving)
    - MediaPlayer.framework
    - StoreKit.framework (Mark as Optional)
    - SystemConfiguration.framework
    - UIKit.framework
    - Security.framework
    - libz.dylib

Likely these frameworks are already incorporated into your project.

CHECKPOINT

Verify that your Project Navigator in the Xcode incorporates Flurry as follows

MoPubSampleApp
3 targets, iOS SDK 8.2
▶ MoPubSDK
▼ AdNetworkSupport
  ▼ Flurry
    h FlurryBannerCustomEvent.h
    m FlurryBannerCustomEvent.m
    h FlurryInterstitialCustomEvent.h
    m FlurryInterstitialCustomEvent.m
    h FlurryMPConfig.h
    m FlurryMPConfig.m
    h FlurryNativeAdAdapter.h
    m FlurryNativeAdAdapter.m
    h FlurryNativeCustomEvent.h
    m FlurryNativeCustomEvent.m
  ▼ SDK
    ▼ Flurry
      h Flurry.h
        libFlurry_6.3.0.a
    ▼ FlurryAds
      ▼ FlurryAdBanner
        h FlurryAdBanner.h
        h FlurryAdBannerDelegate.h
      ▼ FlurryAdInterstitial
        h FlurryAdInterstitial.h
        h FlurryAdInterstitialDelegate.h
      ▼ FlurryAdNative
        h FlurryAdNative.h
        h FlurryAdNativeAsset.h
        h FlurryAdNativeDelegate.h
        h FlurryAdNativeStyle.h
        libFlurryAds_6.3.0.a
      ▼ Shared
        h FlurryAdError.h
        h FlurryAdTargeting.h
      ▼ StaticAPI
        h FlurryAdDelegate.h
        h FlurryAds.h

2. Configure Flurry Ad space(s)

For each MoPub ad unit that you would like to mediate Flurry through, please create a matching ad space on Flurry's dev portal (http://dev.flurry.com).

Log into the developer portal and navigate to the **Publishers** tab. On the left hand navigation bar select **Inventory** and then **Ad Spaces**.
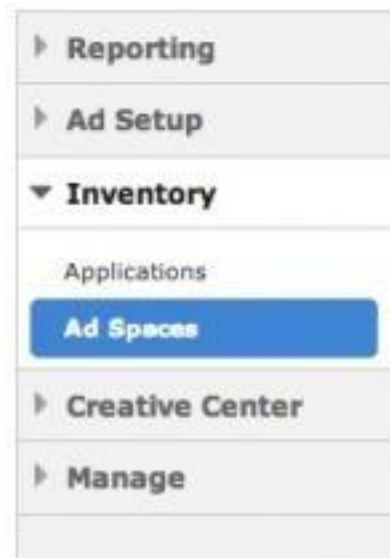
With Ad Spaces selected you'll see an index of previously created ad spaces. To set up a new one, Click on the New Ad Space button on the top right. The Ad Space setup screen has 4 modules.

The Basic Setup section includes fields required to define the name, application, dimensions, placement and orientation of the ad space.

The basic setup is all you need to start, you can click Save.

Please note that mediating Flurry through MoPub requires no additional Flurry related coding. The Flurry ad space invocation code is already incorporated in the FlurryBannerCustomEvent, FlurryInterstitialCustomEvent and FlurryNativeCustomEvent code (added to your project in the previous step).

If you are integrating banner ads, and would like to streamline reconciliation of the impressions count, we recommend you turn off banner refresh on the Flurry side and let MoPub refresh the ads. *This can be done by setting the refresh rate for the ad space to a unusually high number, like 9999 (instead of the default 30 seconds). This setting is found 4 section of the ad space setup - Advanced Options. Change the value for 'Ad Refresh Rate' setting.*

3. Configure MoPub to mediate Flurry

Flurry's custom events are implemented in accordance with instructions provided by MoPub
(https://github.com/mopub/mopub-ios-sdk/wiki/Custom-Events).

After you incorporate the FlurryBannerCustomEvent, FlurryInterstitialCustomEvent and/or
FlurryNativeCustomEvent files into your project as described in the Step 1 of this guide, you
need to configure Flurry as the Custom Native Network into your mediation flow. Please follow
instructions provided by MoPub
https://github.com/mopub/mopub-ios-sdk/wiki/Getting-Started#adding-third-party-ad-networks.

Please note that Flurry Custom Event's classes are called FlurryBannerCustomEvent,
FlurryInterstitialCustomEvent and FlurryNativeCustomEvent, and that you need to provide the
name of Flurry's ad space (as created in the Step 2 of this guide) in the Data (Options)
section of the Custom Event for each ad unit you are mediating. The name of the parameter is
"adSpaceName". (for example, {"adSpaceName": "MoPubBanner"}.

Fullscreen Ad	Full	No Custom Content Entered	FlurryInterstitialCustomEvent	{"adSpaceName": "MoPubTakeover"}
Leaderboard Ad	Leaderboard	No Custom Content Entered	FlurryBannerCustomEvent	{"adSpaceName": "MoPubiPadBanner"}
Flurry Native Stream	Native	No Custom Content Entered	FlurryNativeCustomEvent	{"adSpaceName": "native_ad"}
Banner Ad	Banner	No Custom Content Entered	FlurryBannerCustomEvent	{"adSpaceName": "MoPubBanner"}