Ad Mediation Unity3D Plugin

Note: The online documentation is available here.

How it works

Most mobile publishers have more inventory than they can sell through one ad network or their direct sales team. To maximize fill rates and sell more of their inventory, most publishers work with multiple ad networks at the same time. Ad network performance can also vary for a given publisher, with some offering better performance in a particular geography or industry than another. This means some ad networks will produce more revenue for a publisher than others.

With ad mediation, publishers can match the right ad networks to the right inventory and monetize more of their impressions. Mediation increases fill rates, maximizes eCPMs, and helps publishers get the most revenue possible from every impression, making it a critical monetization solution for publishers.

How Mobile Ad Mediation Works

Ad mediation is technology that sends ad requests to multiple ad networks to ensure publishers find the best available network to fill their ad slots. First, publishers rank ad networks in order of preference. Second, the mediation platform tries the top ad network. If the top ad network can’t fill the ad request, the mediation platform tries the next preferred ad network until it fills the request.

For each ad request, the mediation platform chooses the ad network that best matches the publisher’s priorities. Often, publishers will choose the highest revenue option and prioritize the ad network that offers the highest eCPM. Alternatively, a publisher may give preference to a specific ad network for ad requests coming from that ad network’s country of origin.

They key technical benefit of ad mediation is the ability to centralize access to ad networks with just one SDK, instead of managing hundreds of SDK integrations directly. The single SDK approach allows publishers to add and remove ad networks through the ad mediation platform, instead of pushing out an app update to all app stores. However, you will need to install the Ad SKD's in your project anyways. But instead to work with it's API directly you can manage your game ads thrum Ad Mediation API.

Follow the Setup Guide if you are ready to get started

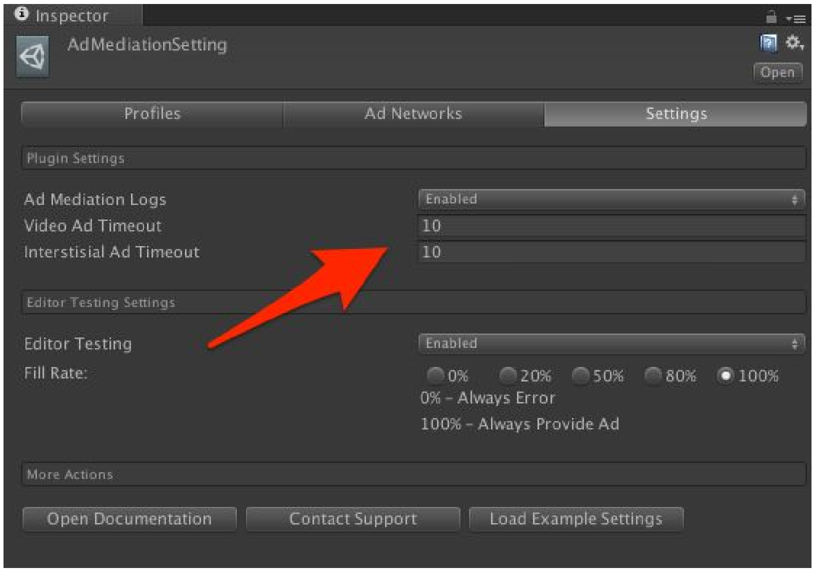
Getting Started

Setup

Ad Mediation provides one single interface for Ad implementation in your game. There is no any specific action required to setup Ad Mediation API. However, before you can use mediation you need to import AD SDK's in your project, each SDK may required some setup action. You can fine setup guide for each SDK's under the Advertising Networks section.

Mediation Profiles

The priority and ad networks is defined by the Mediation profile. When you requesting the ad, mediation API will request the network with highest priority. In case the network failed to provide ad or request waiting time was to high, the Ad Mediation will switch to another network in the scope. If all network failed to load the ad, you will get failed event. You can also set the maximum ad request time in the Ad Mediation settings. If Ad network will not provide ad in specified time, mediation will send the request to another network. The default time is 10 sec. See the screenshot bellow:



Plugin Settings

Ad Mediation Logs: Enabled
Video Ad Timeout: 10
Interstisial Ad Timeout: 10

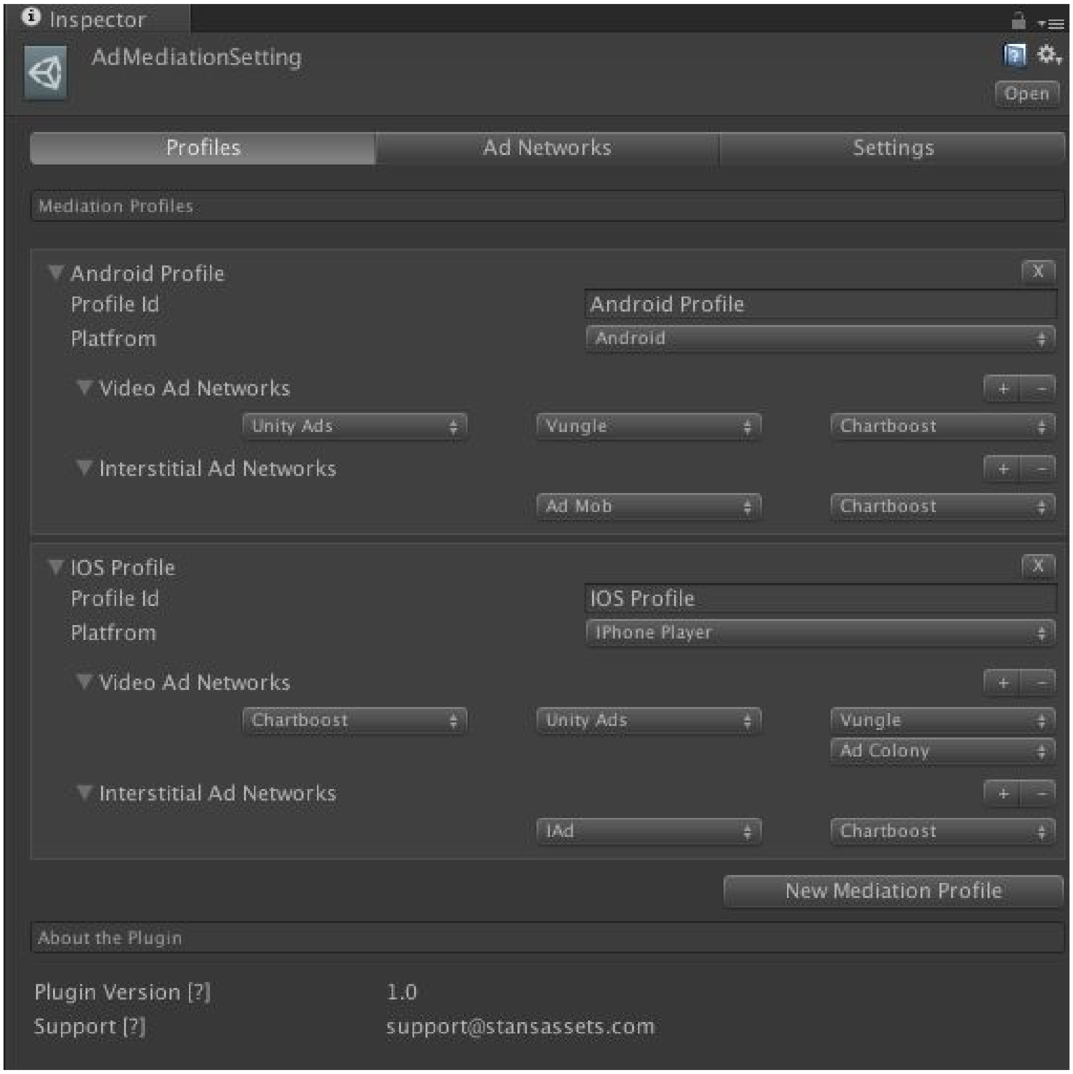
Editor Testing Settings

Editor Testing: Enabled
Fill Rate: ○ 0% ○ 20% ○ 50% ○ 80% ● 100%
0% – Always Error
100% – Always Provide Ad

More Actions

There is few option how you can create and setup the medication profile.

1) You can define the mediation profile using editor Settings, you can also set profiles per platform. As showed on the screenshot bellow.



Mediation Profiles

Android Profile
Profile Id: Android Profile
Platfrom: Android
Video Ad Networks: Unity Ads, Vungle, Chartboost
Interstitial Ad Networks: Ad Mob, Chartboost

IOS Profile
Profile Id: IOS Profile
Platfrom: IPhone Player
Video Ad Networks: Chartboost, Unity Ads, Vungle, Ad Colony
Interstitial Ad Networks: IAd, Chartboost

About the Plugin

Plugin Version [?]: 1.0
Support [?]: support@stansassets.com

2) You can Setup mediation profile using C#, it is very useful, if for example you want to set the profile relying not only on current platform but also on different factors like for example country code, etc. Following code snipped demonstrate the same platforms set up as on previous screenshot but using the C# implementation.

```
M_MediationProfile p = new M_MediationProfile();

switch(Application.platform) {
case RuntimePlatform.Android:
        p.VideoProviders.Add(M_VideoProvider.UnityAds);
        p.VideoProviders.Add(M_VideoProvider.Vungle);
        p.VideoProviders.Add(M_VideoProvider.Chartboost);

        p.InterstitialProviders.Add(M_InterstitialProvider.AdMob);
        p.InterstitialProviders.Add(M_InterstitialProvider.Chartboost);

        break;
case RuntimePlatform.IPhonePlayer:

        p.VideoProviders.Add(M_VideoProvider.Chartboost);
        p.VideoProviders.Add(M_VideoProvider.UnityAds);
        p.VideoProviders.Add(M_VideoProvider.Vungle);
        p.VideoProviders.Add(M_VideoProvider.AdColony);

        p.InterstitialProviders.Add(M_InterstitialProvider.iAd);
        p.InterstitialProviders.Add(M_InterstitialProvider.Chartboost);

        break;
}

M_Settings.Instance.Profiles.Add(p);
AdMediation.Instance.Init ();
```

Note: If you want to add mediation profiles using C# you need to do this before the **AdMediation** Initialization.

Continue to the Implementation guide to find out how to use the mediation API in your game.

Implementation

Maker sure you have Setup the Mediation Profile, before starting to use mediation API.

First of all you need to initialize it. Basically initialization will simply pick mediation profile, so there is no initialization callback, just call Init method on your game start.

```
AdMediation.Instance.Init ();
```

Interstitial

You need to load Interstitial before you can load this. As soon the interstitial is loaded you will get OnVideoLoadComplete callback. After Interstitial is loaded you can show it anytime you want.

The sample bellow will show how to load interstitial and show it as soon as it's loaded.

```
AdMediation.Instance.OnVideoLoadComplete += HandleOnVideoLoadComplete;
AdMediation.Instance.LoadInterstitial();

private void HandleOnInterstitialLoadComplete (M_InterstitialLoadResult res) {
        Debug.Log("Ad Mediation Example: Interstitial Loaded: " + res.AdProviderNetwork + " " +
res.IsSucceeded);
        if(res.IsSucceeded) {
                AdMediation.Instance.ShowInterstitial();
        } else {
                Debug.Log("Interstitial Loading failed");
        }
}
```

You can also findout if interstisial ready to be displayed using **IsInterstitialReady** property

```
void FixedUpdate() {
        if(AdMediation.Instance.IsInterstitialReady) {
                AdMediation.Instance.ShowInterstitial();
        }
}
```

There **OnInterstitialFinished** callback will be fired when interstisial is finished. So if you paused your game after **ShowInterstitial** method call, you can use **OnInterstitialFinished** callback to unpause it.

```
AdMediation.Instance.OnInterstitialFinished += HandleOnInterstitialFinished;
AdMediation.Instance.ShowInterstitial();

private void HandleOnInterstitialFinished (M_InterstitialFinishResult res) {
        Debug.Log("Ad Mediation Example: Interstitial Finished: " + res.AdProviderNetwork + " " 
+ res.IsSucceeded);
}
```

Video Ad

Loading and displaying video ad is pretty much the same as for interstitial.

```
AdMediation.Instance.OnVideoLoadComplete += HandleOnVideoLoadComplete;
AdMediation.Instance.LoadVideo();

private void HandleOnVideoLoadComplete (M_VideoLoadResult res) {
        Debug.Log("Ad Mediation Example: Video Loaded: " + res.AdProviderNetwork + " " + 
res.IsSucceeded);
        if(res.IsSucceeded) {
                AdMediation.Instance.ShowVideo();
        }else {
                Debug.Log("Video Loading failed");
        }
}
```

The **IsVideoReady** property can be used as alternatie to the **OnVideoLoadComplete** callback

```
void FixedUpdate() {
        if(AdMediation.Instance.IsVideoReady) {
                AdMediation.Instance.ShowVideo();
        }
}
```

Some video Ad netroks allow user to skip Video before it will be finished. So analytzing the **M_VideoFinishResult**witch is comes with the **OnVideoFinished** will let you know if you should reward your user for watching the video ad.

```
AdMediation.Instance.OnVideoFinished += HandleOnVideoFinished;
AdMediation.Instance.LoadVideo();

private void HandleOnVideoFinished (M_VideoFinishResult res) {
    Debug.Log("Ad Mediation Example: Video Finished: " + res.AdProviderNetwork + " " +
res.IsSucceeded);
    if(res.IsSucceeded) {
        //Reward user for watching
    }
}
```

Testing in Editor

Ad Mediation allows you to test your implementation in editor, in fact if Editor testing is enabled Ad Meadion SDK will show the Test Ad UI, simulating the behavior on a real device. You can enable or disable the Editor Testing feature with the plugin settings, and also set the Ad fill rate for testing.

In case Editor Testing will be disable calls to **AdMediation** API inside the Unity Editor will be simply ignored.

More use examples can be found under the **UnifiedAdsTestScene** Scene.

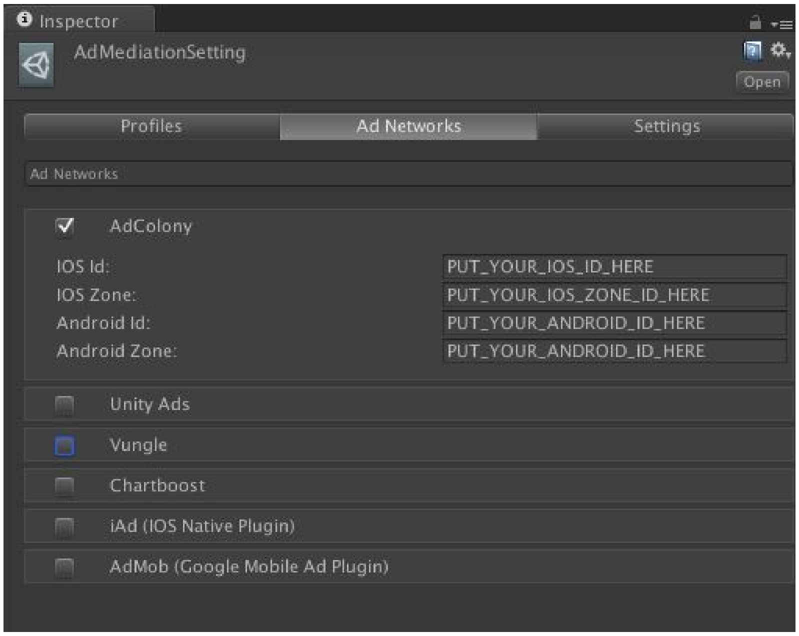
Advertising Networks

AdColony

Before you can enable AdColony Network, make sure that you already have the AdColony account. Otherwise, point your browser to the AdColony website and create one.

After you done with the registration, create your app using AdColony Dashboard, and import AdColony Unity SDKinto your project.

Now you ready to enable AdColony Mediation.
Open plugin settings at:
Window -> Stan's Assets -> Ad Mediation -> Edit Settings
Go to the Ad Networks tab and fill AdColony setting after enabling it. As showed on the screenshot below.



☑ AdColony

IOS Id: PUT_YOUR_IOS_ID_HERE
IOS Zone: PUT_YOUR_IOS_ZONE_ID_HERE
Android Id: PUT_YOUR_ANDROID_ID_HERE
Android Zone: PUT_YOUR_ANDROID_ID_HERE

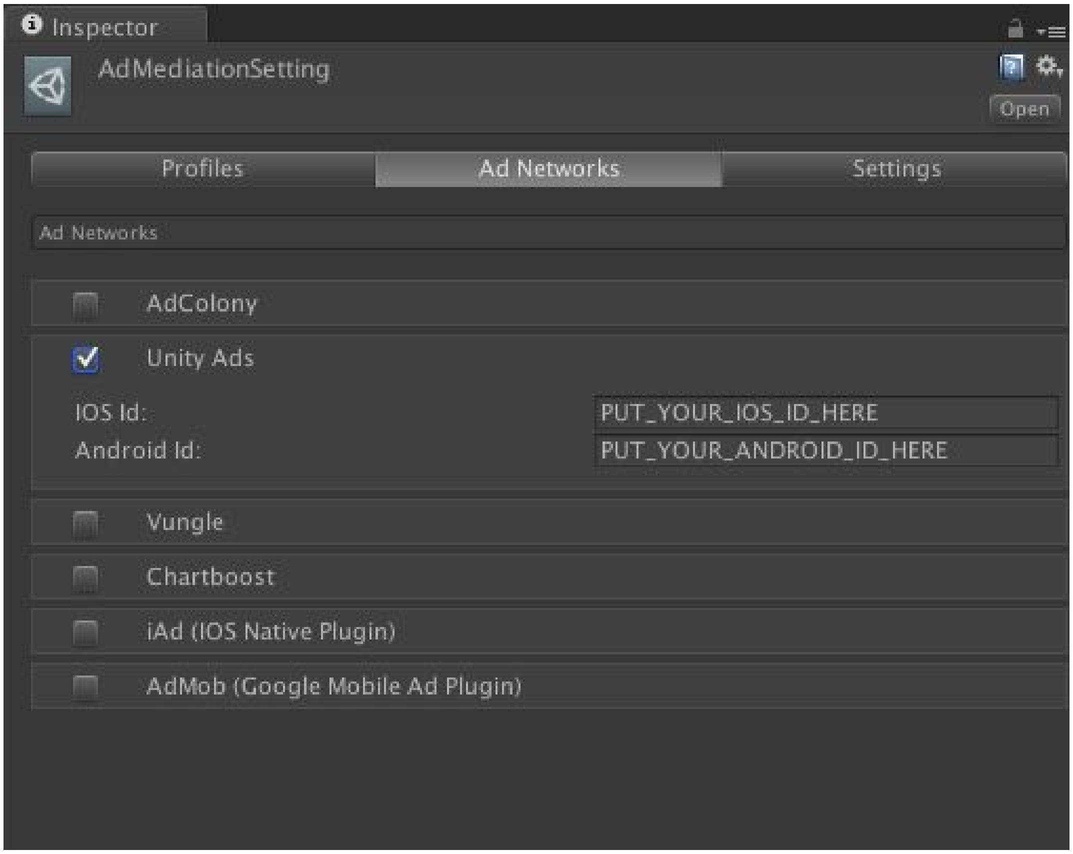
☐ Unity Ads
☐ Vungle
☐ Chartboost
☐ iAd (IOS Native Plugin)
☐ AdMob (Google Mobile Ad Plugin)

Once you done with the Setup, you may use the AdColony Network in your Mediation Profile.

Unity Ads

Enabling Unity Ads is fairly simple, and takes just 3 steps.

- Grab Unity Ads SDK from the Asset Store.
- Point your browser to the Unity Ads Service Dashboard and add your app by following the onscreen instructions.
- Open plugin settings at *Window -> Stan's Assets -> Ad Mediation -> Edit Settings*, enable and fill up the Unity Ads Network setting as showed on the screenshot bellow.



Ad Networks

□ AdColony

☑ Unity Ads

IOS Id: PUT_YOUR_IOS_ID_HERE

Android Id: PUT_YOUR_ANDROID_ID_HERE

□ Vungle

□ Chartboost

□ iAd (IOS Native Plugin)

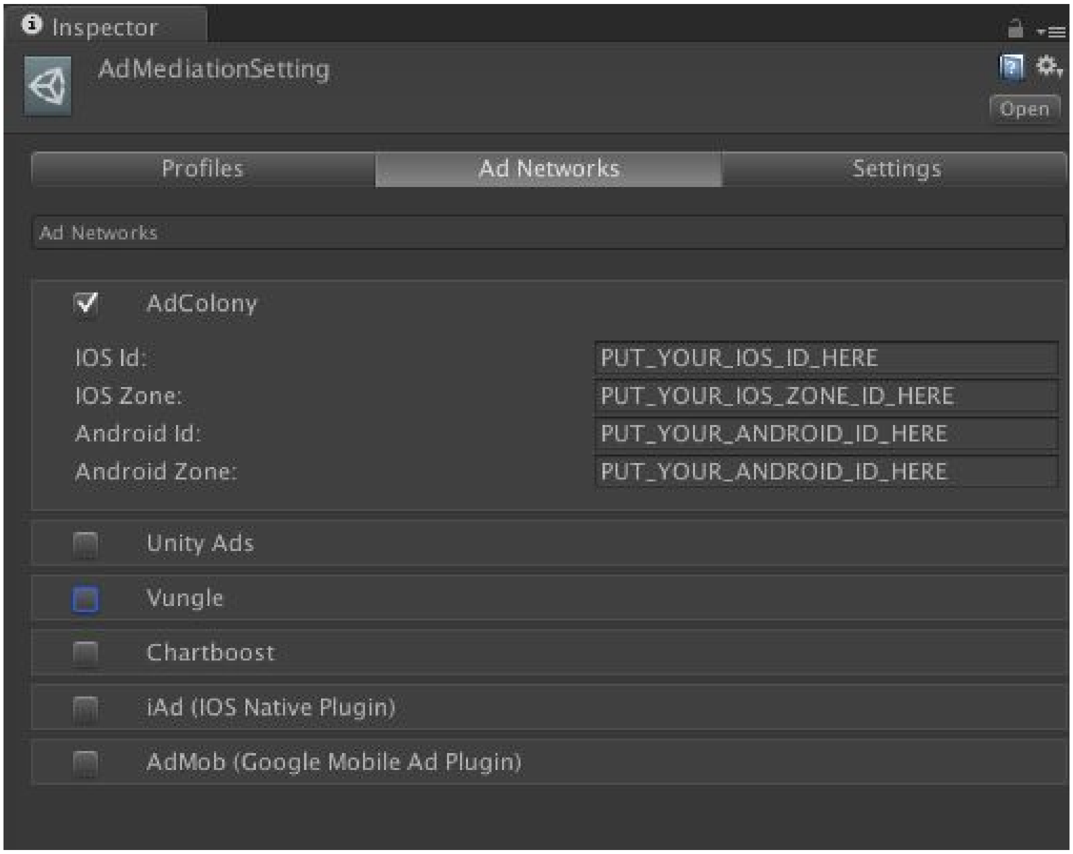
□ AdMob (Google Mobile Ad Plugin)

Once you done with the Setup, you may use the Unity Ads Network in your Mediation Profile.

Vungle

Setting up Vungle SDK is pretty much the same. Just follow few simple steps.

- Register your app using Vungle Dashboard at vungle.com
- Import Vungle Unity SDK inside your project.
- Enable and fill up the Vungle Network settings as showed on the screenshot bellow.



AdMediationSetting

Ad Networks

☑ AdColony

IOS Id: PUT_YOUR_IOS_ID_HERE
IOS Zone: PUT_YOUR_IOS_ZONE_ID_HERE
Android Id: PUT_YOUR_ANDROID_ID_HERE
Android Zone: PUT_YOUR_ANDROID_ID_HERE

☐ Unity Ads

☐ Vungle

☐ Chartboost

☐ iAd (IOS Native Plugin)

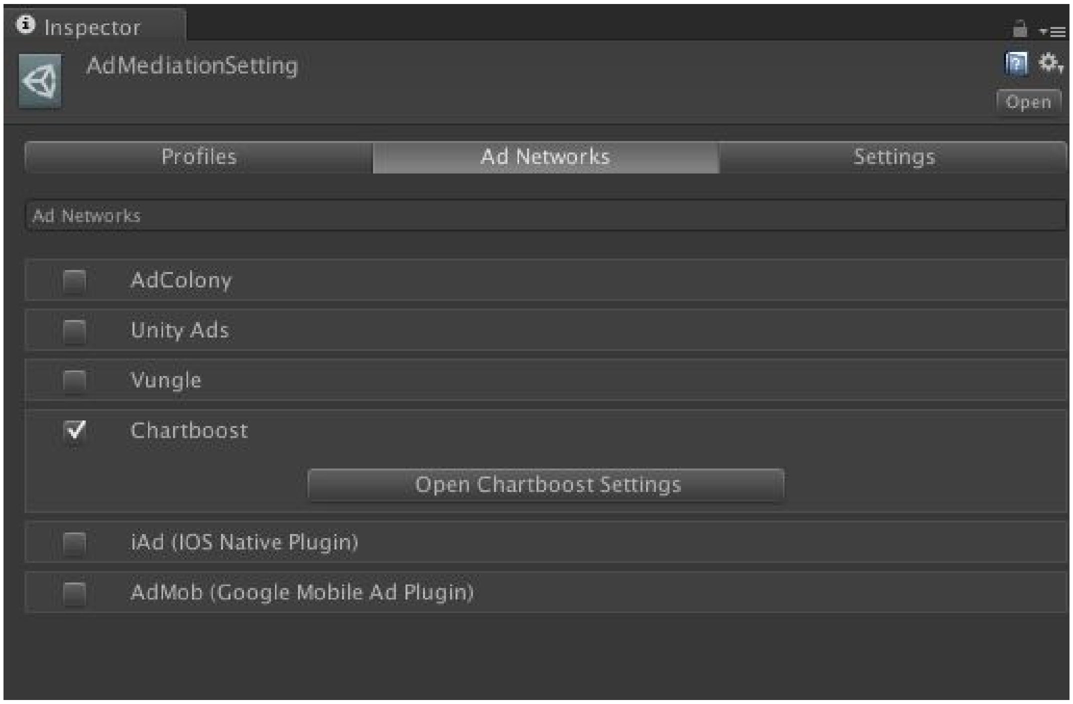
☐ AdMob (Google Mobile Ad Plugin)

Once you done with the Setup, you may use the Vungle Network in your Mediation Profile.

Chartboost

Flow is absolutely the same as for previous Networks.

- Create account and register your app at chartboost.com
- Import Chartboost Unity SDK inside your project.
- Enable and fill up the Chartboost Network settings as showed on the screenshot bellow.



☐ AdColony
☐ Unity Ads
☐ Vungle
☑ Chartboost
☐ iAd (IOS Native Plugin)
☐ AdMob (Google Mobile Ad Plugin)

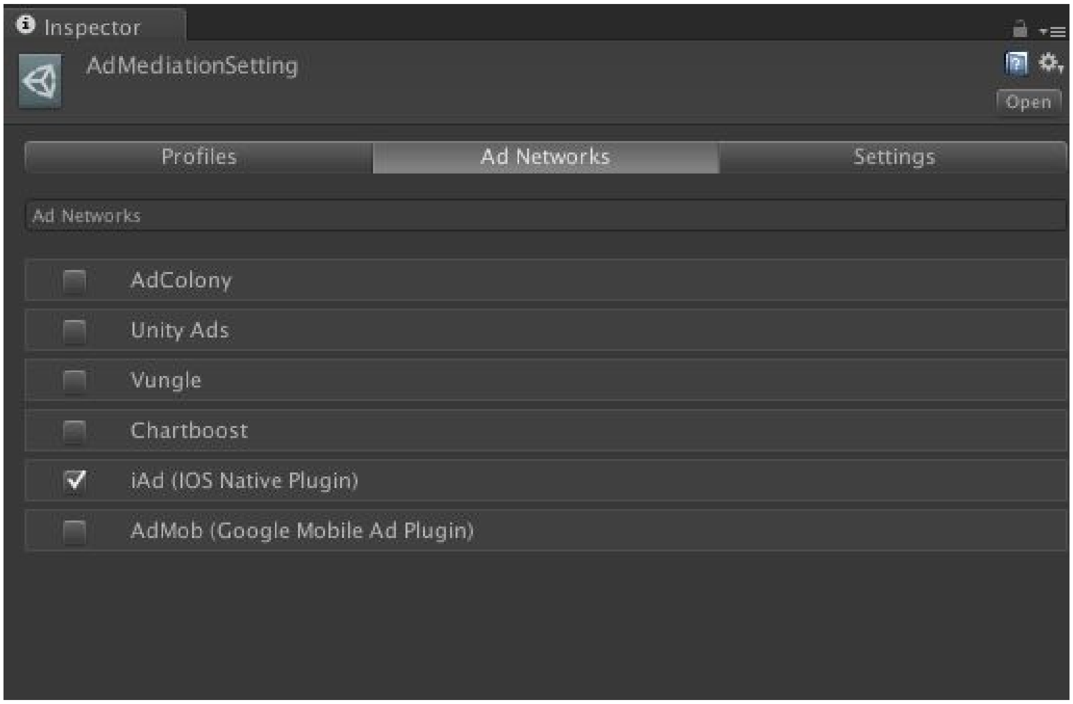
Once you done with the Setup, you may use the Chartboost Network in your Mediation Profile.

iAd

Ad Mediation Plugins is using iAd implementation from the IOS Native plugins which is part of Stan's Assets products. In order to enable iAd Netwok you need to own IOS Native copy, which can be purchased from the Asset Store.

So set up steps would be:

- Import IOS Native plugin to your project.
- Setup iAd in iTunes connect
- Enable iAd Network in plugin settings as showed on the screenshot bellow.



AdMediationSetting

Ad Networks

☐ AdColony
☐ Unity Ads
☐ Vungle
☐ Chartboost
☑ iAd (IOS Native Plugin)
☐ AdMob (Google Mobile Ad Plugin)

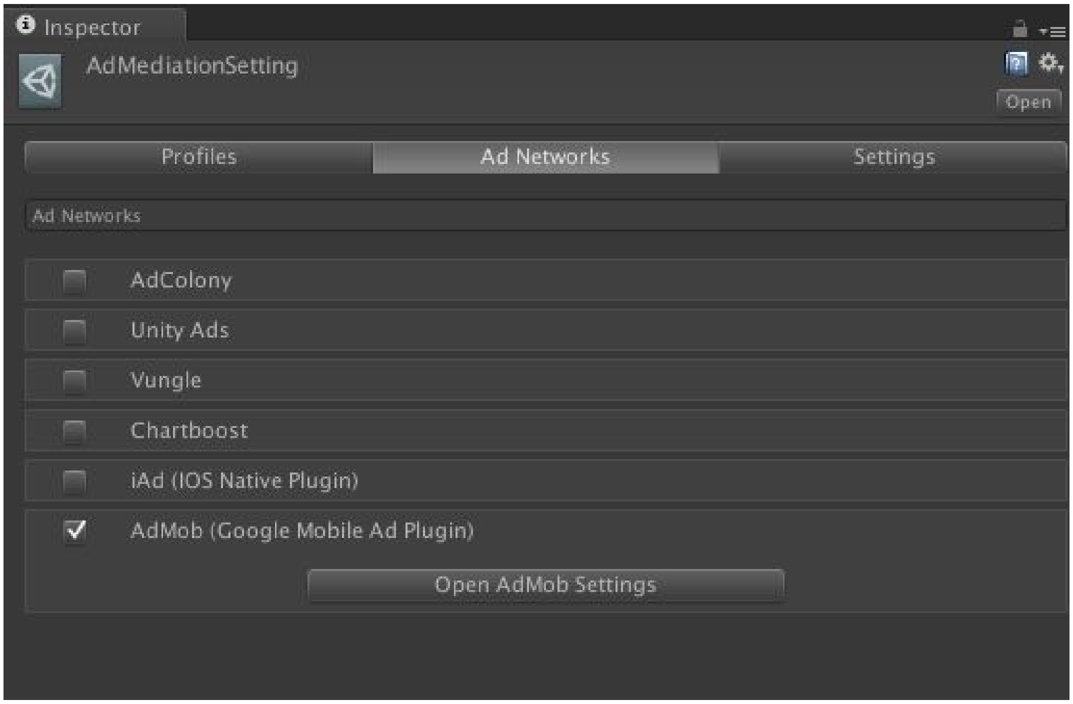
Once you done with the Setup, you may use the iAd Network in your Mediation Profile.

AdMob

Ad Mediation Plugins is using Admob implementation from the Google Mobile Ad plugin which is part of Stan's Assets products. In order to enable Admob Network you need to ownGoogle Mobile Ad plugin copy, which can be purchased from the Asset Store.

So set up steps would be:

- Import Google Mobile Ad plugin to your project.
- Setup the Admob account.
- Enable AdMob Network in plugin settings as showed on the screenshot bellow.



Ad Networks

- □ AdColony
- □ Unity Ads
- □ Vungle
- □ Chartboost
- □ iAd (IOS Native Plugin)
- ☑ AdMob (Google Mobile Ad Plugin)

Once you done with the Setup, you may use AdMob Network in your Mediation Profile.