

Freestar Ads Mediation Xamarin iOS

Last Modified on 05/31/2022 10:41 am EDT

Freestar Ads Mediation provides support for Xamarin iOS in C# by providing the necessary NuGet package.

[Change Log](#)

Supported Ad Partners

Note: The supported list for Xamarin may differ from our Native iOS

Ad Provider	SDK Version	Ad Unit Types
AdColony	4.1.4	Fullscreen Interstitial & Rewarded
Amazon	3.0.0	Fullscreen Interstitial, Banner 300x250, Banner 320x50
AppLovin	6.11.5	Fullscreen Interstitial & Rewarded, Banner 320x50
Criteo	3.4.2	Fullscreen Interstitial, Banner 300x250, Banner 320x50
Admob	7.53.1	Fullscreen Interstitial & Rewarded, Banner 300x250, Banner 320x50
Google Ads Manager	7.53.1	Fullscreen Interstitial & Rewarded, Banner 300x250, Banner 320x50
Unity Ads	3.4.2	Fullscreen Interstitial & Rewarded, Banner 320x50

Getting Started

Start displaying Freestar Ads in your Xamarin app today by following the simple steps below.

Requirements

- Before we begin, you must have a working Xamarin app running on an iOS device. This document will not show how to create a Xamarin app for iOS as that would be beyond the scope of Freestar Ads Mediation.
- Your Xamarin app must target `Xamarin.iOS 9.0 or higher`.

NuGet Dependencies

In Visual Studio, INSTALL the following NuGet packages:

NuGet Package Name	Version
--------------------	---------

Xamarin.Essentials	1.3.1
--------------------	-------

xam.ios.freestar_ads	3.2.5
----------------------	-------

Info.plist

Some ad networks require adding special parameters in the app's `Info.plist` file. Below are the partner-specific `Info.plist` requirements.

Google Ad Manager

```
<key>GADIsAdManagerApp</key>
<true/>
```

Google Admob

```
<key>GADApplicationIdentifier</key>
<string>{YOUR_ADMOB_KEY}</string>
```

AppLovin

```
<key>AppLovinSdkKey</key>
<string>{YOUR_APPLOVIN_KEY}</string>
```

AdColony

```
<key>NSCalendarsUsageDescription</key>
<string>Adding events</string>
<key>NSPhotoLibraryUsageDescription</key>
<string>Taking selfies</string>
<key>NSCameraUsageDescription</key>
<string>Taking selfies</string>
<key>NSMotionUsageDescription </key>
<string>Interactive ad controls</string>
```

In the above entry, you should change the reasons to be appropriate for your app.

Logging

You can enable detailed logging from the SDK to inspect the ad mediation process in detail via an app's console. This is done by setting a flag in the `Info.plist` file:

```
<key>CHP_LOGGING_ENABLE</key>
<string>true</string>
```

Once this change is made, rebuild the app to see the logs.

⚠ Warning: For both performance and security reasons, it is not advisable to have detailed logging in production apps. Remove the flag in the Info.plist before submitting to the App Store.

Using the Freestar SDK

Note: Any Xamarin class calling Freestar code will need the following directive: `using`

```
FreestarAds;
```

Freestar Initialization

In the `FinishedLaunching` of your starting AppDelegate:

```
Freestar.InitWithAdUnitID("P8RIA3"); //use our test key
```

Interstitial Ads

How to display full screen Interstitial ads:

First, implement the `IFreestarInterstitialDelegate` interface.

Note: How to implement an interface in C# is beyond the scope of this document.

```
FreestarInterstitialAd interstitialAd = new FreestarInterstitialAd([YOUR IFreestarInterstitialDelegate]);  
interstitialAd.LoadPlacement(null); //Note: you may pass a "placement" string instead.
```

If you plan to use more than one placement in your app, please adhere to the placement naming convention as follows:

"my_placement_name_pN", where N is the number of your placement.

For example, let us assume you are using 2 interstitial ad placements in your game or app. The first placement would be the default placement; simply do not specify a placement name by calling the `loadPlacement` method with null as the argument. The second placement would be, for example, "my_search_screen_p1". The ending "p1" tells the SDK to use the second placement you created in our web dashboard for the interstitial ad unit.

This placement format is the same for all the other ad units, such as rewarded ads and banner ads.

When you receive `FreestarInterstitialLoaded` from your `IFreestarInterstitialDelegate`, you may show the ad:

```
interstitialAd.ShowFrom([YOUR App's current top ViewController]);
```

Rewarded Ads

How to display full screen Rewarded ads:

First, implement the `IFreestarRewardedDelegate` interface.

Note: How to implement an interface in C# is beyond the scope of this document.

Choose the reward name and amount, and pass it to the created ad:

```
FreestarReward rwd = FreestarReward.BlankReward();  
rwd.RewardAmount = 100;  
rwd.RewardName = "gems";  
FreestarRewardedAd rewardedAd = new FreestarRewardedAd(YOUR IFreestarRewardedDelegate], rwd);  
  
rewardedAd.LoadPlacement(null); //Note: you may pass a "placement" string instead.
```

When you receive `FreestarRewardedLoaded` from your `IFreestarRewardedDelegate`, you may show the ad:

```
rewardedAd.ShowFrom([YOUR App's current top ViewController]);
```

Banner Ads

How to display Banner ads:

```
FreestarBannerAd bannerAd = new FreestarBannerAd(  
    [YOUR IFreestarBannerAdDelegate],  
    FreestarBannerAdSize.FreestarBanner320x50);  
//For MREC size, use FreestarBannerAdSize.FreestarBanner300x250  
bannerAd.LoadPlacement(null); //Note: you may pass a "placement" string instead.
```

Next, you will receive `FreestarBannerLoaded` of your `IFreestarBannerAdDelegate`. Here is an *example* implementation:

```
public void IFreestarBannerAdDelegate.FreestarBannerLoaded(FreestarBannerAd ad)  
{  
    ad.Center = new CGPoint(  
        YOUR_AdContainer.Frame.Width / 2,  
        YOUR_AdContainer.Frame.Height / 2);  
    YOUR_AdContainer.AddSubview(ad);  
}
```

Sample Reference App

Finally, we have a [Sample Reference Xamarin iOS app](#). Please use for reference purposes only as the Visual Studio project might not build or compile on your local machine.

Important iOS 14 Changes

One thing you must do as a publisher is set the Facebook `setAdvertiserTrackingEnabled` flag appropriately, since Facebook is one of our partners. [Here is the official document from Facebook](#). Generally speaking, one typical you could approach this is if the user has granted the app permission to track them at the OS level, then you would set this flag to true. By default, the flag will be false, meaning that Facebook will not serve any ads. We leave this implementation detail to you in order to give you, the app publisher, more control.

GDPR Update

We are releasing a major update to the Freestar SDK later this week that will require publishers to utilize a GDPR TCF 2.0 compliant CMP in order to render ads to users in a GDPR affected country. If the publisher is not utilizing a GDPR TCF 2.0 compliant CMP once upgraded to this SDK version, then ads will not serve to users in a GDPR affected country.

We put together a comprehensive list of FAQs below to help break down how the changes may impact your business. If you have additional questions, please reach out to your dedicated Account Manager and we will be happy to help on a case by case basis.

What is the new SDK version?

4.0.0 for iOS and Android

How do I become GDPR compliant?

In order to be able to serve ads to your users who reside in any EU country, you will need to implement an IAB TCF 2.0 CMP service.

The CMP service is essentially a personal data and privacy form that must be presented to users to collect their consent or dissent. This form can be presented as often as you like throughout the session of your game or app, as you may require your users to see ads. The most typical setup would be for the first visit to an App with an option to edit your preferences through some other call to action.

What if I already have a CMP implemented?

If you already have a CMP implemented, you will need to add Publisher First, Inc. (Freestar's official legal entity) as a vendor, as well as each of our partners listed here if not already included.

You will need to retrigger the consent form for GDPR affected users once Freestar and additional vendors are added to the vendor list.

If the list of vendors has already been included, no additional effort is required.

How does the Freestar SDK work with a CMP?

Freestar SDK will automatically detect the user's CMP response and act accordingly. More specifically, if the user consents, then Freestar SDK will be allowed to serve ads. If the user dissents, then Freestar SDK will not show ads.

What if I choose not to implement a CMP?

If a CMP is not implemented at all, then Freestar SDK will not show ads to users who reside in a GDPR affected country.

What do I need to do once I choose a CMP?

After you have chosen a CMP Service provider, during configuration and setup of your CMP, you will need to include a list of supported vendors. We recommend that you select 'Include All Vendors' rather than choosing specific vendors, as your list of vendors may change over time.

If you choose to add vendors to your list manually, you will need to include Publisher First, Inc. (Freestar's official legal entity) as a vendor as well as Freestar's list of supported vendors here.

Does Great Britain still observe GDPR requirements despite leaving the E.U.?

Yes, it does still apply. In anticipation of Brexit, a new domestic data privacy law called the UK-GDPR took effect on 1/31/20. The UK-GDPR is almost word for word completely identical to the EU's GDPR.

CMP Recommendations:

Consent Manager - <https://www.consentmanager.net/> App Consent - <https://sfbx.io/en/produits/>

Here is a list of IAB approved CMP Service providers you can implement in your game or app: <https://iab europe.eu/cmp-list/>

Testing

For iOS, please use our iOS test key **91784edd-3492-4111-8742-f71bd3803dd3** (we have a different key for Android, so don't use for both iOS and Android) for all your iOS testing runs and enable test mode.

Turn on test mode:

```
FSTR_TEST_ADS
```

You will usually get 100% fill on all ad units.

It is not recommended to use your production key for testing runs as that is strictly prohibited by our partners and bad things may happen to us on the business side of things.

Do not forget to uninstall and re-install your app when changing keys on your device.

When you are satisfied with your testing, please make a release build with your production key, and turn test mode off. Publish to store.

SKAdNetwork IDs

iOS 14 changed the way advertising works on iOS devices. Ad effectiveness tracking requires the usage of `SKAdNetwork` APIs. To enable this, `SKAdNetwork` keys should be included in your app's `Info.plist` file.

Please see our list: https://github.com/freestarcapital/SDK_documentation_iOS/wiki/iOS-14-SKAdNetwork-IDs

M1 Macs

Currently, we do not support the building of our SDK on M1 macs. This will be addressed in an upcoming release.

Xcode 12 related cocoapods build errors

Recently, in Xcode 12.2, Apple made some breaking changes in Xcode related to build settings. By default, arm64 architecture is now added to ARCHS_STANDARD. In addition, they have removed support for VALID_ARCHS setting from Xcode. Apple also added, in Xcode 12, a new build setting called **Excluded Architectures**. The reason Apple added arm64 is to support simulator on M1 Mac hardware. However this causes build issues running simulator on Intel Macs. As a result, publishers running Xcode on Intel Macs will need to make usage of the EXCLUDED_ARCHS build setting, for simulator testing, to exclude the arm64 simulator. So, if your Xcode build is failing with arm64 simulator errors in the log, such as:

```
building for iOS Simulator, but linking in object file built for iOS, file for architecture arm64
```

then, it is suggested to use this cocoapods post install hook in your Podfile:

```
post_install do |installer|
  installer.pods_project.build_configurations.each do |config|
    config.build_settings["EXCLUDED_ARCHS[sdk=iphonesimulator*]"] = "arm64"
  end
end
```

This is a temporary fix and should address the issue of build failures resulting from Xcode attempting to build (arm64) simulator on Intel Macs. As of our latest SDK release, we do this automatically in the podspec via xcconfigs, however not every SDK vendor has adopted this

workaround. This post install hook is only needed if you are running Xcode 12 on Intel hardware and wanting to run simulator, and you have a cocoapods dependency that is not Xcode 12 / arm64 compliant.

Firebase Dependency

If your app integrated the Firebase SDK, reference the Firebase version compatible with current release 7.0

```
pod 'Firebase/Core', '~> 7.0'
```

For further reference:

<https://github.com/CocoaPods/CocoaPods/issues/10104>

<https://stackoverflow.com/questions/63607158/xcode-12-building-for-ios-simulator-but-linking-in-object-file-built-for-ios/63955114#63955114>

Yahoo Demand Partner Configuration

If Yahoo is being added as a demand partner in your Podfile, then it is required to add an entry into your Info.plist. See example below:

```
<key>VerizonAdsSourceAppId</key>  
<string>Replace_this_with_your_App's_App_Store_ID</string>
```

GAM and Googleleadmob OB adapters

If you would like to make usage of Google Open Bidding (OB) adapters for FreestarAds, please contact your account manager (AM) for further instructions. We do not advise including OB adapters into your project without first consulting with your AM.