

Freestar Ads Mediation Xamarin Android

Last Modified on 05/31/2022 12:00 pm EDT

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

Supported Ad Partners

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

Ad Provider	SDK Version	Ad Unit Types
AdColony	4.1.1	Fullscreen Interstitial & Rewarded
Amazon	5.9.0	Fullscreen Interstitial, Banner 300x250, Banner 320x50
AppLovin	9.11.4	Fullscreen Interstitial & Rewarded, Banner 320x50
Criteo	3.4.0	Fullscreen Interstitial, Banner 300x250, Banner 320x50
Admob	17.2.2	Fullscreen Interstitial & Rewarded, Banner 300x250, Banner 320x50
Google Ads Manager	17.2.2	Fullscreen Interstitial & Rewarded, Banner 300x250, Banner 320x50
Facebook Ads	5.8.0	Fullscreen Interstitial & Rewarded, Banner 300x250, Banner 320x50
Unity Ads	3.4.0	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 Android device. This document will not show how to create a Xamarin app for Android as that would be beyond the scope of Freestar Ads Mediation.
- Your Xamarin app must target `MonoAndroid 9.0 or higher`.

NuGet Dependencies

In Visual Studio, INSTALL the following NuGet packages:

NuGet Package Name	Version
Xamarin.Android.Support.Compat	28.0.0.3
Xamarin.Android.Support.Core.UI	28.0.0.3
Xamarin.Android.Support.Fragment	28.0.0.3
Xamarin.Android.Support.Annotations	28.0.0.3
Xamarin.Android.Support.VersionedParcelable	28.0.0.3
Xamarin.Android.Support.v4	28.0.0.3
Xamarin.Android.Support.Media.Compat	28.0.0.3
Xamarin.GooglePlayServices.Ads	71.1720.1
Xamarin.Facebook.AudienceNetwork.Android	5.+
xam.android.freestar_ads	3.2.2.2
xam.android.freestar_adcolony_adapter	4.1.2.1
xam.android.freestar_admob_adapter	17.2.0.1
xam.android.freestar_amazon_adapter	5.9.0
xam.android.freestar_applovin_adapter	9.11.6.2
xam.android.freestar_gam_adapter	17.2.0.1
xam.android.freestar_criteo_adapter	3.4.0
xam.android.freestar_facebook_adapter	5.8.0.1
xam.android.freestar_unityads_adapter	3.4.0

AndroidManifest.xml

Please add the following meta-data tags to the `Application` tag of the `AndroidManifest.xml` *Note: The Google and AppLovin values below are non-production values.*

```
<meta-data
  android:name="com.google.android.gms.ads.APPLICATION_ID"
  android:value="ca-app-pub-3940256099942544~3347511713"/>

<meta-data
  android:name="applovin.sdk.key"
  android:value="qwFfGU-34yfW7sQ0boTdLe5GOlb6TG1HAHxRnwUrs8M--o-aZSCv5IAT1naV1bnCuidec6UTFsSEdR
qCv15_JG" />

<meta-data android:name="com.google.android.gms.version" android:value="@integer/google_play_services_verse
ion"/>
```

Freestar Initialization

In the `onCreate` of your starting Activity:

```
FreeStarAds.EnableTestAds(true); //set 'false' for production
FreeStarAds.Init( [YOUR PARENT ACTIVITY], "XqjhRR"); //Use our test key
```

Interstitial Ads

How to display full screen Interstitial ads:

First, implement the `IInterstitialAdListener` interface.

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

```
InterstitialAd interstitialAd = new InterstitialAd( [YOUR Parent Activity] , [YOUR IInterstitialAdListener] );

AdRequest adRequest = new AdRequest( [YOUR Parent Activity] );

interstitialAd.LoadAd(adRequest); //Note: you may pass an optional "placement" string as 2nd parameter if you wish.
```

When you receive `OnInterstitialLoaded` from your `IInterstitialAdListener` , you may show the ad:

```
interstitialAd.Show();
```

Rewarded Ads

How to display full screen Rewarded ads:

First, implement the `IRewardedAdListener` interface.

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

```
RewardedAd rewardedAd = new RewardedAd ( [YOUR Parent Activity] , [YOUR IRewardedAdListener] );

AdRequest adRequest = new AdRequest( [YOUR Parent Activity] );

rewardedAd.LoadAd(adRequest); //Note: you may pass an optional "placement" string as 2nd parameter if you wish.
```

When you receive `OnRewardedVideoLoaded` from your `IRewardedAdListener`, you may show the ad:

```
/*
 Secret: Any arbitrary string you want to identify the reward by. e.g. "12345678". Cannot be blank.
 User Name: Can be the username of the user trying to obtain the reward; may be blank or null.
 Coin: The name of the type of reward. e.g. gold coin; may be blank or null.
 30: The amount of the reward in string. may be blank or null.
*/
rewardedAd.ShowRewardAd("mysecret", "MyUsername", "Gold Coins", "30");
```

Banner Ads

How to display Banner ads:

```
AdRequest adRequest = new AdRequest( [YOUR Parent Activity] );
BannerAd bannerAd = new BannerAd( [YOUR PARENT ACTIVITY] );
bannerAd.AdSize = AdSize.Banner32050; //For MREC size, use AdSize.MediumRectangle300250
bannerAd.SetBannerAdListener( [YOUR IBannerAdListener] );
bannerAd.LoadAd(adRequest); //Note: you may pass an optional "placement" string as 2nd parameter if you wish.
```

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

```
public void OnBannerAdLoaded(View bannerAdView, string placement)
{
    BannerAd ad = (BannerAd)bannerAdView;
    if (ad.AdSize == AdSize.Banner32050)
    {
        YOUR_AdContainer.RemoveAllViews();
        YOUR_AdContainer.AddView(bannerAdView); //FreeStar ad will be displayed now
    }
}
```

Sample Reference App

Please see the `Xamarin_Android_Sample` subfolder under [this repo](#)

Open using Microsoft Visual Studio

Privacy - Google Play Families Policy Compliance

If your game or app is officially under the [Google Play Families](#) program, Freestar provides such support:

```
FreeStarAds.setGoogleFamilyPolicyMode( GoogleFamilyPolicyMode.app, true); //If your app is designed only for children
//FreeStarAds.setGoogleFamilyPolicyMode( GoogleFamilyPolicyMode.mixed, false); //If your app is designed for families with children
FreeStarAds.init(...);
```

If your app is not officially under the Google Play Families program, then you do not need to set the Google Family Policy mode.

```
/**
 * Only set Google Families Policy Mode if your app is required to comply with Google Play's
 * Families Policy program.
 *
 * @param googleFamilyPolicyMode app: the app is child-directed and will not receive
 *                               personalize or contextual ads.
 *                               mixed: the app is directed at mixed audiences.
 *                               none: (default) the app is not required to comply with Google
 *                               Play's Family Policy
 *
 * @param onlyNonPersonalizedAds true: if 'mixed' mode, then only personalized or contextual ads
 *                               may be served.
 *                               false: if 'mixed' mode, then personalized or contextual ads
 *                               may be served.
 *                               note: if 'app' mode, then personalized or contextual ads
 *                               may not be served regardless of this parameter.
 */
public static void setGoogleFamilyPolicyMode(GoogleFamilyPolicyMode googleFamilyPolicyMode,
                                             boolean onlyNonPersonalizedAds)
```

GDPR

Freestar SDK is GDPR compliant. In order for your users to be able to receive any ad fills in GDPR-affected countries, you, as a publisher, will need to implement a 3rd party Consent Management Platform (CMP). Freestar SDK will automatically detect user interactions with the CMP and act accordingly. Please see our [Freestar GDPR Frequently Asked Questions](#) for complete details and our recommended list of CMP service providers.

Testing

For Android, please use our test key `XqjhRR` for all your testing runs and enable test mode true. 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.

Automated Testing - Bypassing Ads

Are your automated tests failing after integrating Freestar Mediation Ads into your mobile application or game? Are you not sure it could be due to Freestar or something else? We have a feature called **Automated Test Mode** where you can run your automated tests to bypass Freestar or run Freestar in 'Limited Mediation' mode without making drastic changes to your code:

In your automated test suite code, **before** `FreeStarAds.init` is called:

```
FreeStarAds.setAutomatedTestMode( FreeStarAds.AutomatedTestMode.BYPASS_ALL_ADS )  
//OR  
FreeStarAds.setAutomatedTestMode( FreeStarAds.AutomatedTestMode.LIMITED_MEDIATION ) //only runs AdMob &  
GAM
```

Again, this is only for your automated tests and not for production use.

Release Build

Final note when creating your release build: if you use [proguard](#) or [minify](#), please add our [proguard rules](#) to your app proguard file.