

Mobile Ads Documentation

To insure a quick response to any issues with the asset please send all support requests to the following e-mail address:

support@bizzybeegames.com

Please include the asset name and Unity version you are using. Thank you!

Table of Contents:

Mobile Ads Documentation	1
Ads	2
AdMob Setup	2
Unity Ads Setup	6
Consent	10
Setting the Consent Status	10
Troubleshooting	11

Ads

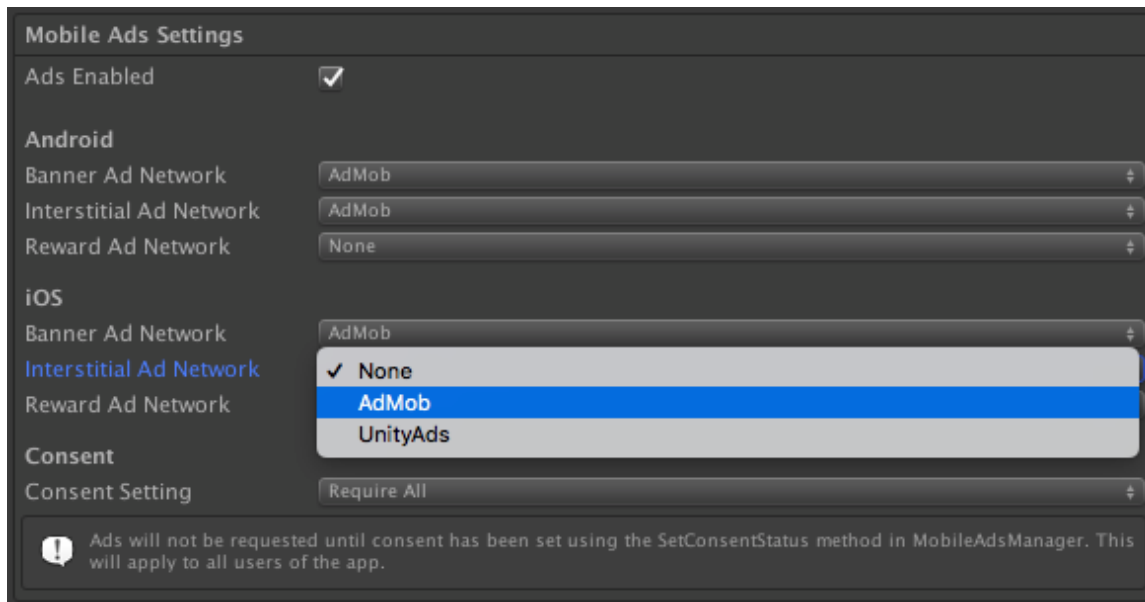
Ads are setup using the **Mobile Ads Settings** window which can be opened by selecting the menu item **Tools -> Bizzy Bee Games -> Mobile Ads Settings** (Or clicking the button on the MobileAdsManager inspector).

On the Mobile Ads Settings window you can select either **AdMob** or **Unity Ads** to be used for Banner, Interstitial, and/or Reward ads for both Android and iOS platforms. Selecting **None** will disable ads for that platform / ad type.

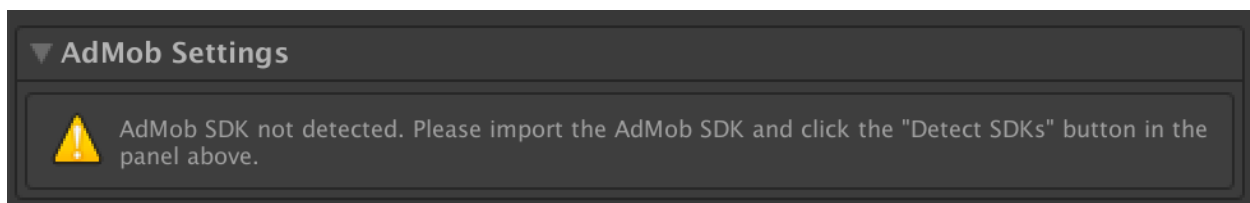
NO FILL ERROR - If you have setup AdMob and you are getting the **no fill** error message when trying to load ads it means the AdMob server does not have any available ad inventory to give your application. This usually happens with a new AdMob account / app. After a couple days your app should start receiving ads. This error is **not** an implementation error and **cannot** be solved with any code changes. If after 7 days your app is still not receiving ads you will have to contact AdMob support to rectify the issue.

AdMob Setup

Step1. Select AdMob in one or more of the drop downs.




A new section will appear at the bottom of the window called **AdMob Settings**. Expanding it now will display the following warning:



Step2. Download and import the AdMob Unity SDK by clicking on this link <https://github.com/googleads/googleads-mobile-unity/releases> and clicking the GoogleMobileAds.unitypackage

Download the GoogleMobileAds.unitypackage:

Google Mobile Ads Unity Plugin v3.18




 stowy released this on Jun 20 · 21 commits to master since this release

- Added GoogleMobileAdsSettings editor UI for making Plist / manifest changes.
- Fix OnRewardedAdFailedToShow callbacks.
- Migrated android support library to androidx (JetPack) with Google Mobile Ads SDK version 18.0.0.

Built and tested with:

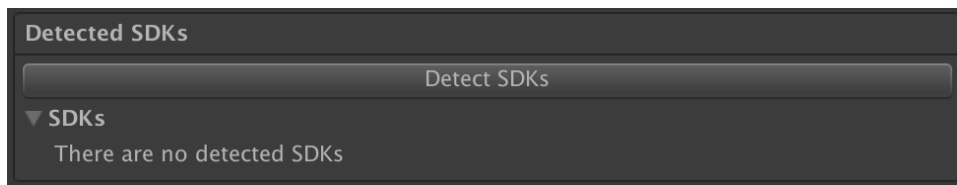
- Google Play services 18.0.0
- Google Mobile Ads iOS SDK 7.45.0
- Unity Jar Resolver 1.2.119

▼ Assets 3

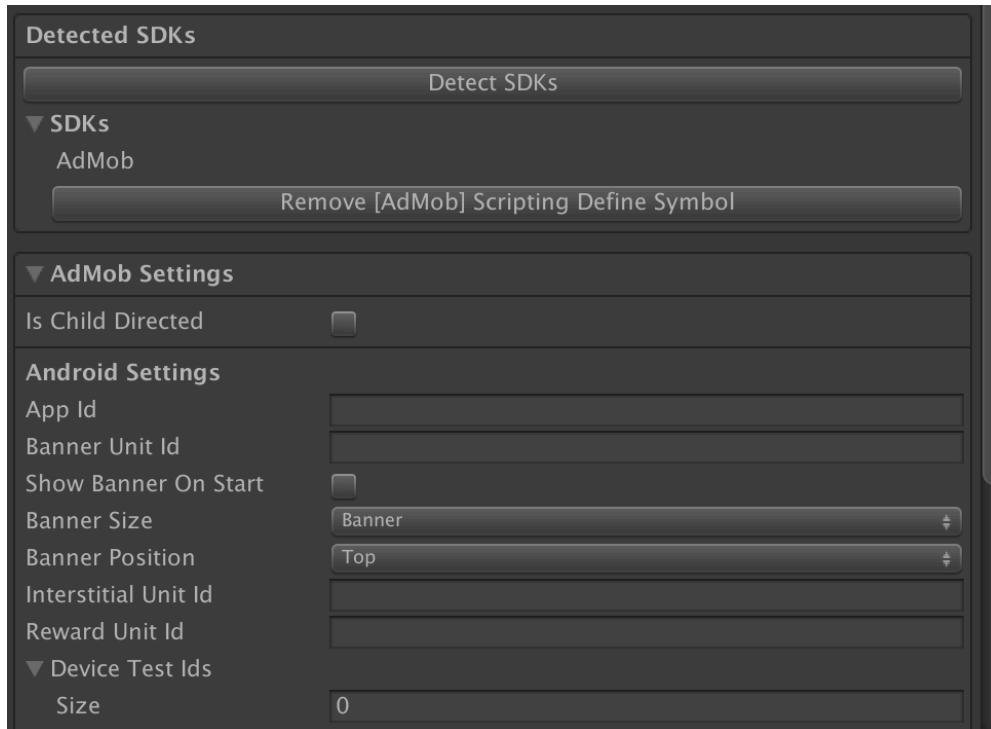
	GoogleMobileAds-v3.18.0.unitypackage	385 KB
	Source code (zip)	
	Source code (tar.gz)	

Step3. Once the GoogleMobileAds.unitypackge has finished importing into Unity click the Detect SDKs button on the Mobile Ads Settings window:

NOTE: This button will not work if there are any compiler errors in your project. If there are compiler errors they will have to be resolved first before clicking the Detect SDKs button.

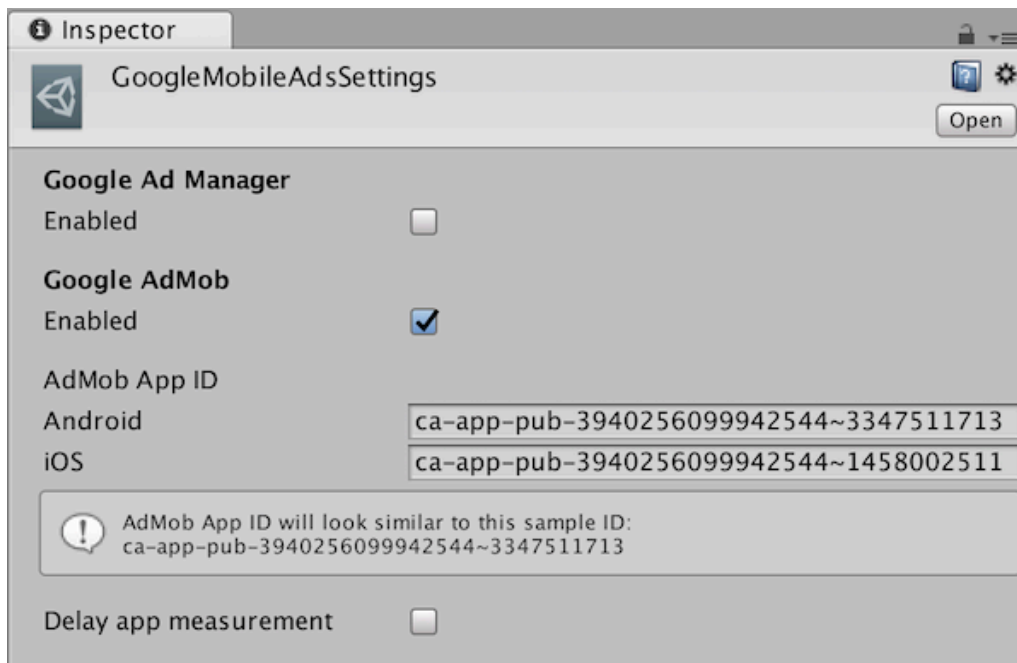


After Unity finishes compiling AdMob should appear under the SDKs list and the AdMob fields should appear under AdMob Settings



Step4. Add your App Id and Unit Ids to the fields under AdMob Settings

Step5. Select **Assets > Google Mobile Ads > Settings** from the menu. Enable AdMob by clicking Enabled checkbox under Google AdMob section. Then enter your Android and iOS AdMob app ID in each field.

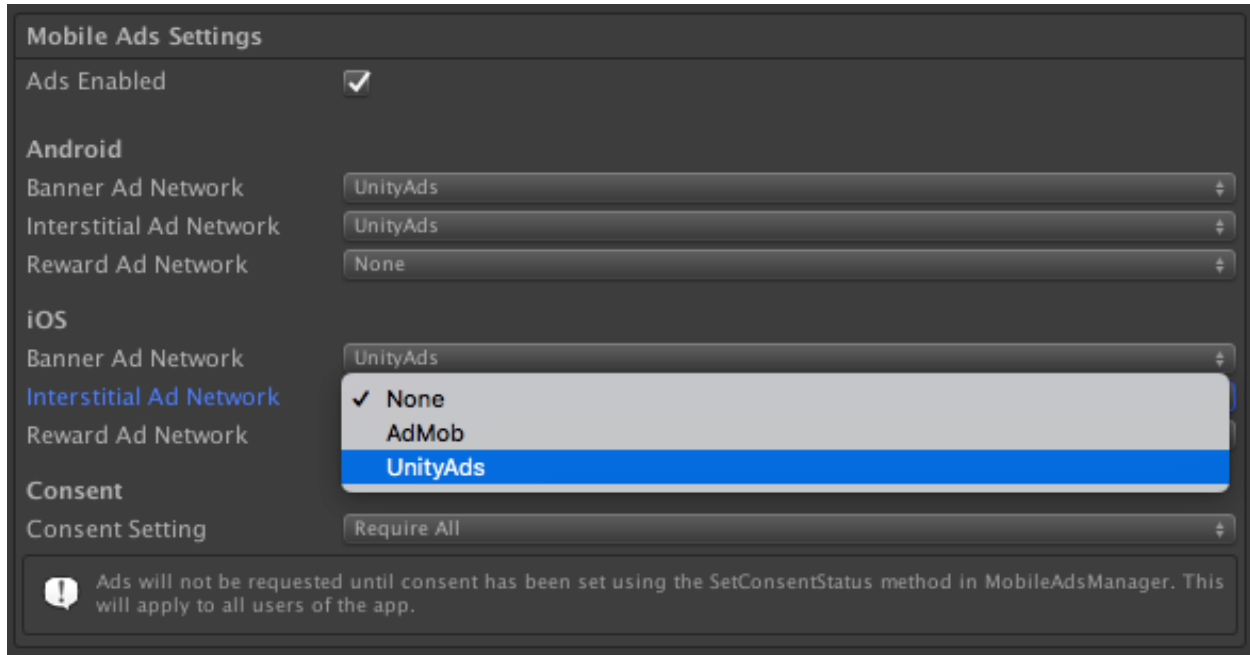


Step6 [Android]. Make sure the Play Services Resolver has run by selecting the menu item **Assets -> Play Services Resolver -> Android Resolver -> Resolve.**

Step6 [iOS]. Make sure Cocoapods has been installed by selecting the menu item **Assets -> Play Services Resolver -> iOS Resolver -> Install Cocoapods.**

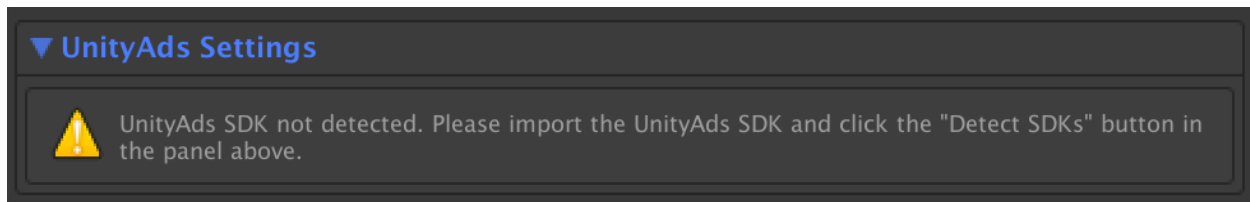
Unity Ads Setup

Step1. Select Unity Ads in one or more of the drop downs.



The screenshot shows the 'Mobile Ads Settings' panel. It has a dark grey background with white text. At the top, 'Mobile Ads Settings' is written in a bold font. Below it, 'Ads Enabled' is checked with a white checkmark. Under the 'Android' section, 'Banner Ad Network' is set to 'UnityAds', 'Interstitial Ad Network' is set to 'UnityAds', and 'Reward Ad Network' is set to 'None'. Under the 'iOS' section, 'Banner Ad Network' is set to 'UnityAds', 'Interstitial Ad Network' is set to 'None' (with a checkmark), 'AdMob' is listed below it, and 'UnityAds' is highlighted in blue at the bottom of the dropdown. 'Reward Ad Network' is also present. Under the 'Consent' section, 'Consent Setting' is set to 'Require All'. At the bottom, there is a warning icon and text: 'Ads will not be requested until consent has been set using the SetConsentStatus method in MobileAdsManager. This will apply to all users of the app.'

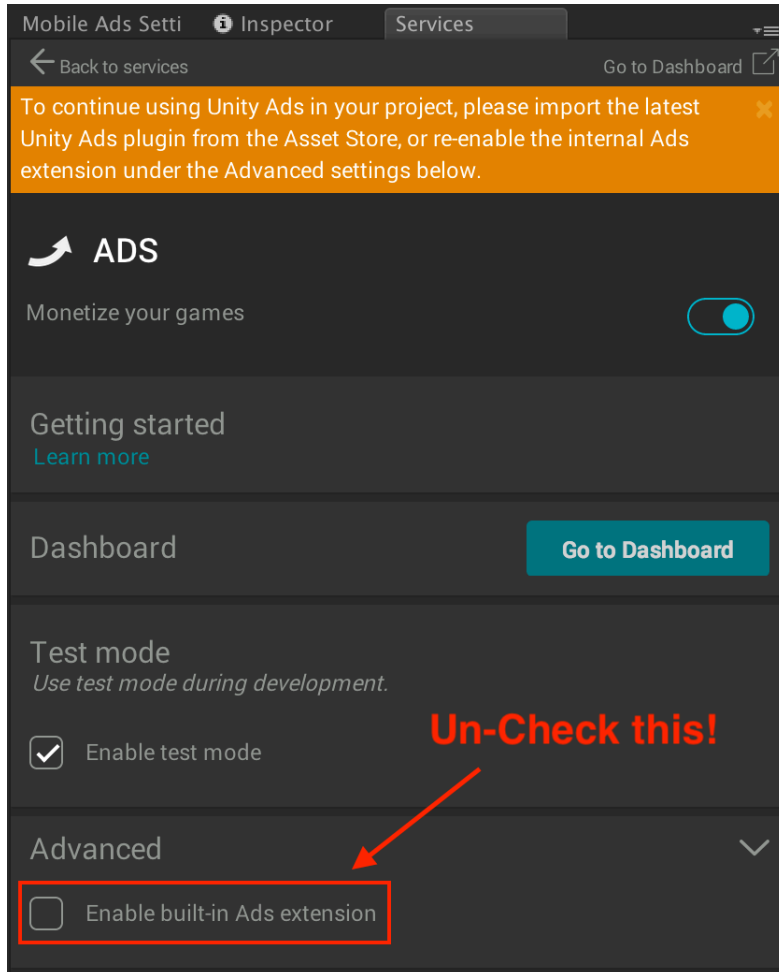
A new section will appear at the bottom of the window called **UnityAds Settings**. Expanding it now will display the following warning:



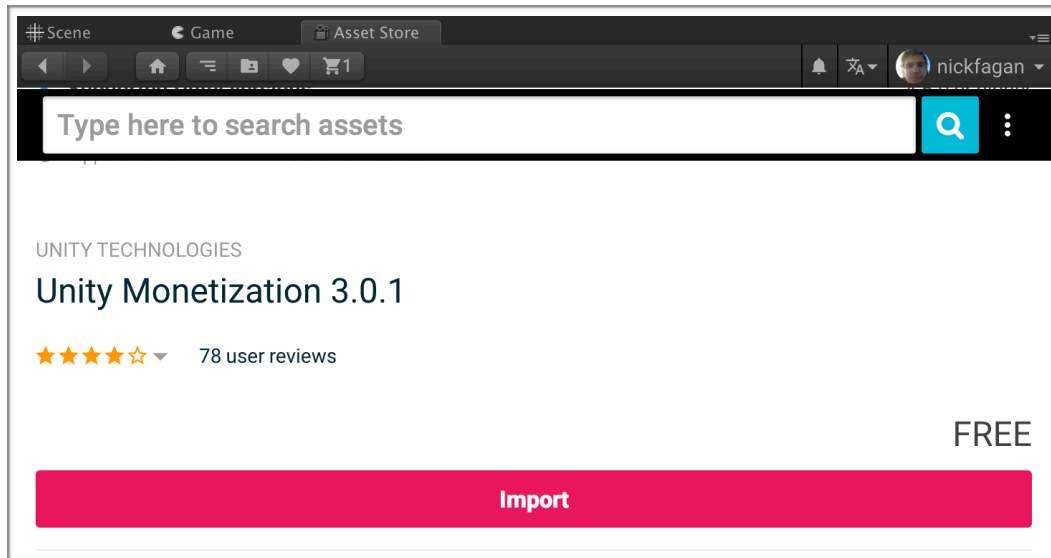
The screenshot shows the 'UnityAds Settings' panel expanded. It has a dark grey background. At the top, '▼ UnityAds Settings' is written in a bold font. Below it, there is a yellow warning icon and text: 'UnityAds SDK not detected. Please import the UnityAds SDK and click the "Detect SDKs" button in the panel above.'

Step2. Enable Ads in the Services window:

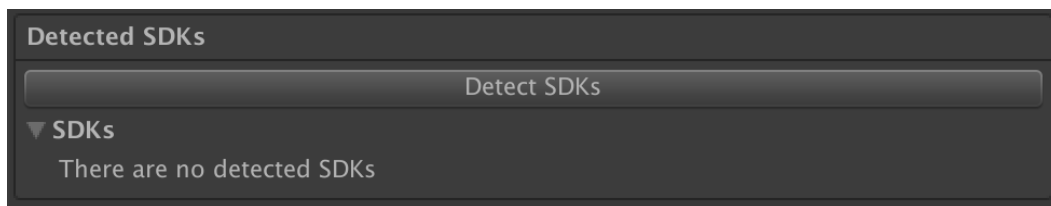
***** IMPORTANT ***** Make sure “Enable built-in Ads Extension” is disabled or it will collide with the Monetization plugin you will import in the next step. To do so expand the **Advanced** section and un-check the field if it is checked



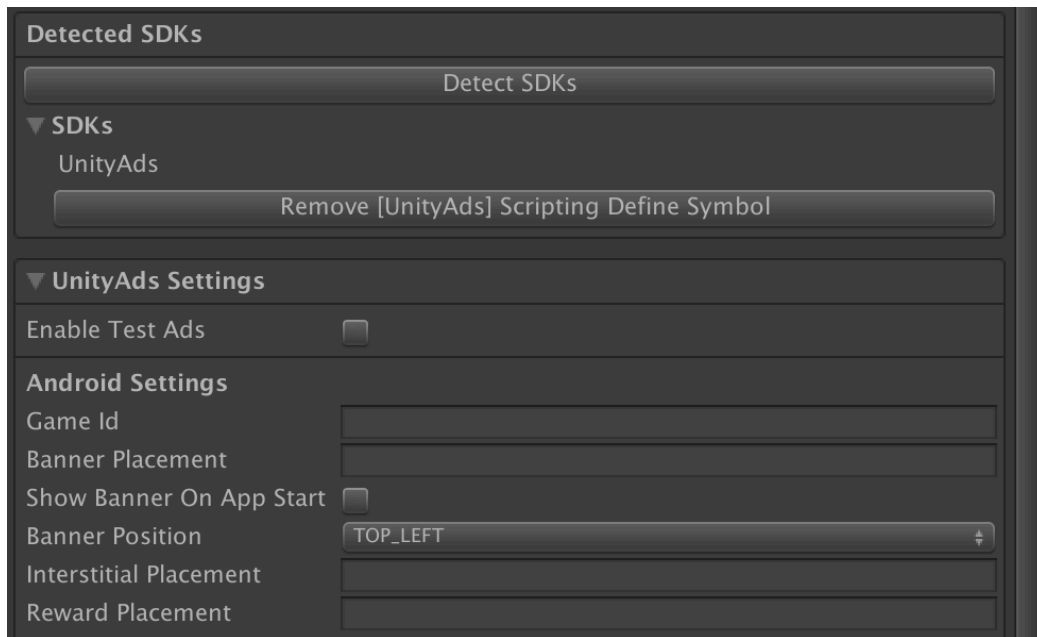
Step3. Open the Asset Store window and Download/Import the Unity Monetization asset:



Step4. Click the **Detect SDKs** button on the Mobile Ads Settings window:



After Unity finishes compiling, UnityAds should appear under the SDKs list and the UnityAds fields should appear under UnityAds Settings:



Step5. Add you Game Id and Placement Ids to the fields under UnityAds Settings.

Thats it! Unity Ads will now appear in your game.

Consent

Consent can be required before any ads are loaded by setting the **Consent Setting** on the Mobile Ads Manager. There are three types you can set the consent setting to:

Not Required - Consent is not required for ads to be loaded.

Required Only In EEA - Consent is only required for users in the European Economic Area. When the app starts it first attempts to determine if the user is located in the EEA and if so ads will not be loaded until the consent status has been set to either Personalized or Non-Personalized. If the user location can not be determined for any reason then it errs on the side of caution and requires consent before ads are loaded.

Require All - Consent is required for all users before ads are loaded.

Setting the Consent Status

If consent is required before ads are loaded then the **SetConsentStatus** method must be called on the MobileAdsManager to set the consent status to either Personalized or Non-Personalized ads.

To set the consent simply call the method like so:

```
MobileAdsManager.Instance.SetConsentStatus(consentStatus);
```

The consentStatus parameter is an integer value:

- 1** - Indicates the user has consented to receive personalized ads
- 0** - Indicates the user should only be shown non-personalized ads.

Troubleshooting

Common problems that could happen:

After adding the AdMob SDK and making an iOS build, Xcode displays the error “library not found for -lGoogleUtilities” or any other Google related library:

You need to open the .xcworkspace file and not the .xcodeproj file. If there is no .xcworkspace file in your build folder then you need to install Cocoapods. (Check Step6 in the AdMob section of this documentation.