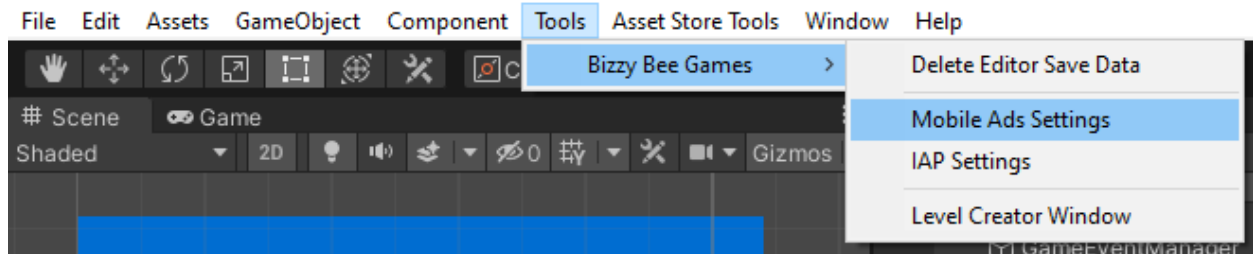


Mobile Ads Documentation

Mobile Ads Settings Window	2
Ad Networks	2
Consent	2
Setting the Consent Status	2
AdMob Setup	4
Step 1	4
Step 2	4
Step 3	5
Step 4	6
Step 5	6
Step 6	7
Unity Ads Setup	8
Step 1	8
Step 2	9
Step 3	10
Step 4	10
Step 6	11
Displaying Ads	12
Test Scene	13

Mobile Ads Settings Window

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**



Ad Networks

The ad networks which are supported are AdMob and Unity Ads. You can set a different ad network for each ad type (interstitial, reward, and banner).

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:

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

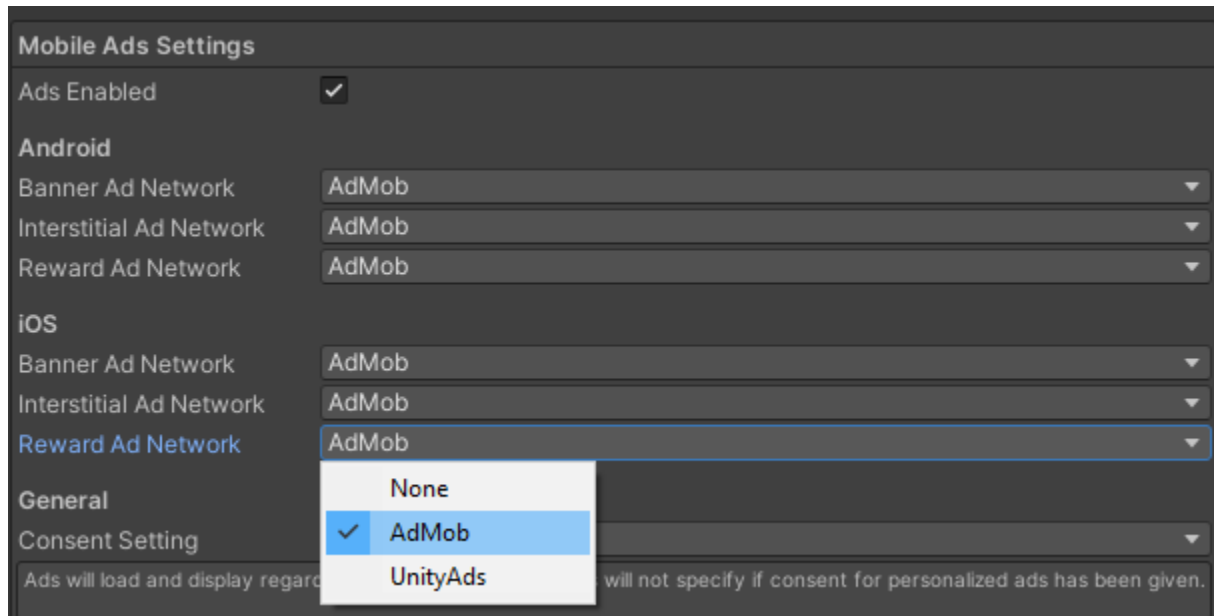
The `consentStatus` parameter is an integer value:

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

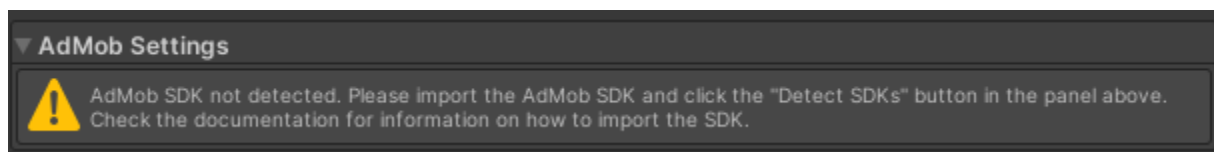
AdMob Setup

Step 1

Select AdMob in one or more of the drop downs.

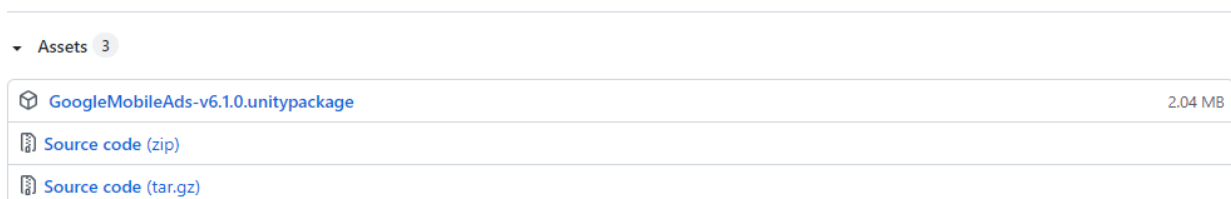


A new section will appear at the bottom of the window called **AdMob Settings**. Expanding it now will display the following warning:

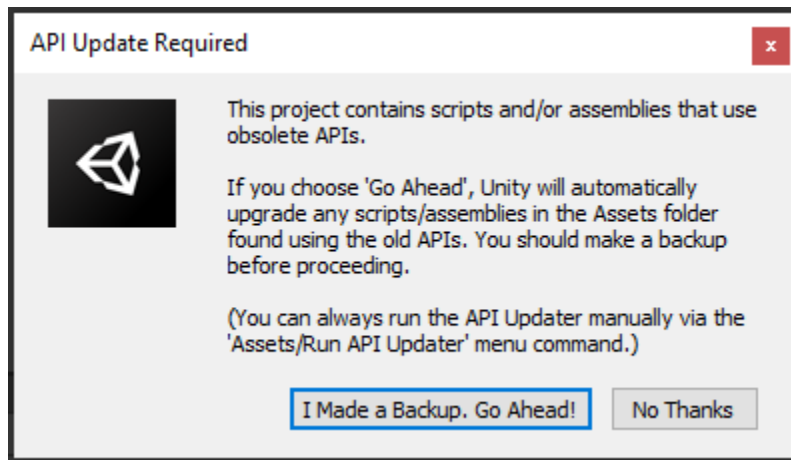


Step 2

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 then importing it into your project.



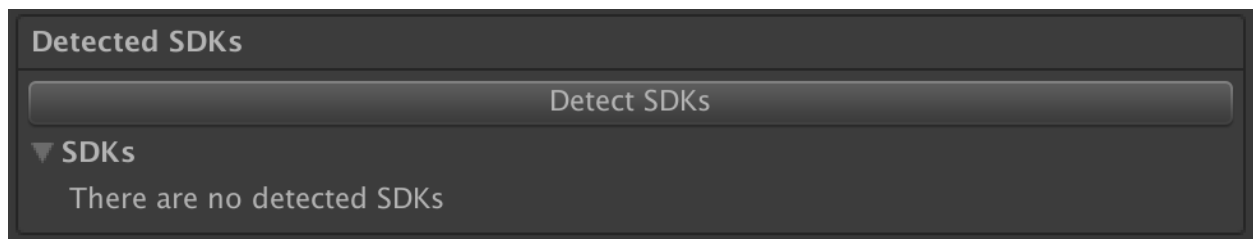
If you get the **API Update Required** dialog you MUST click “I Made a Backup. Go Ahead!” button or AdMob will not import properly.



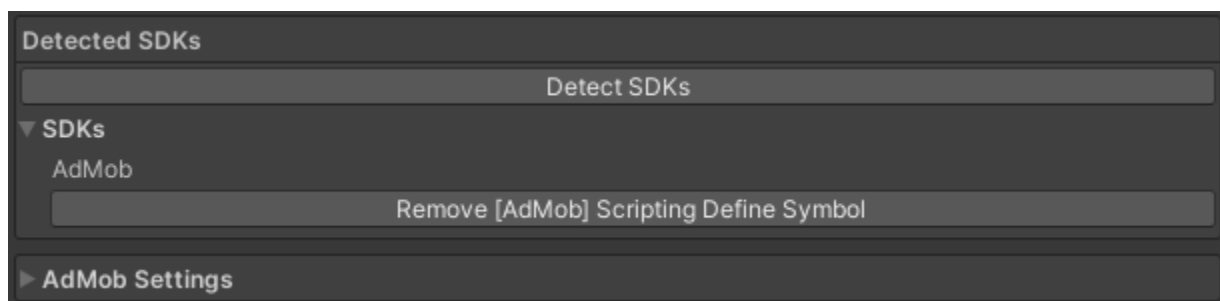
Step 3

Once the GoogleMobileAds.unitypackage 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

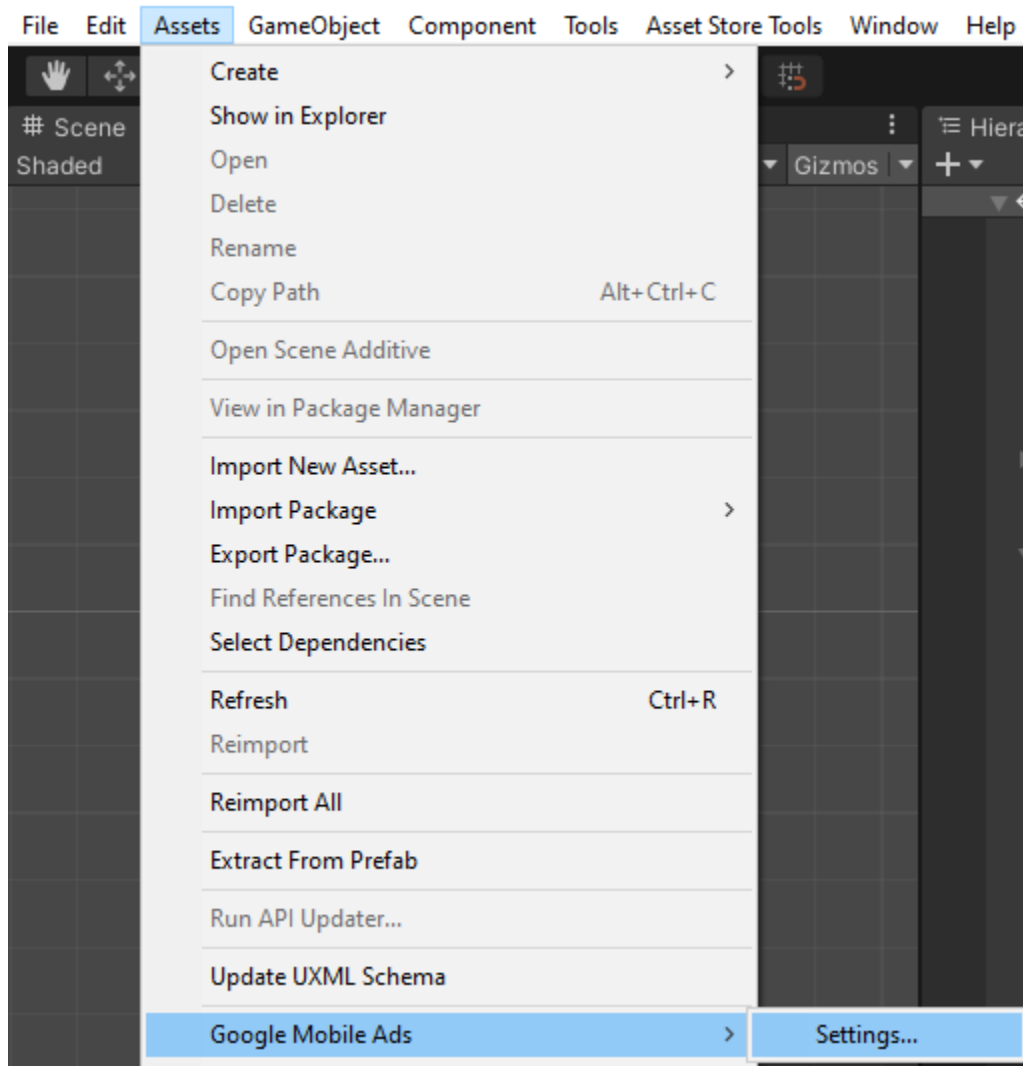


Step 4

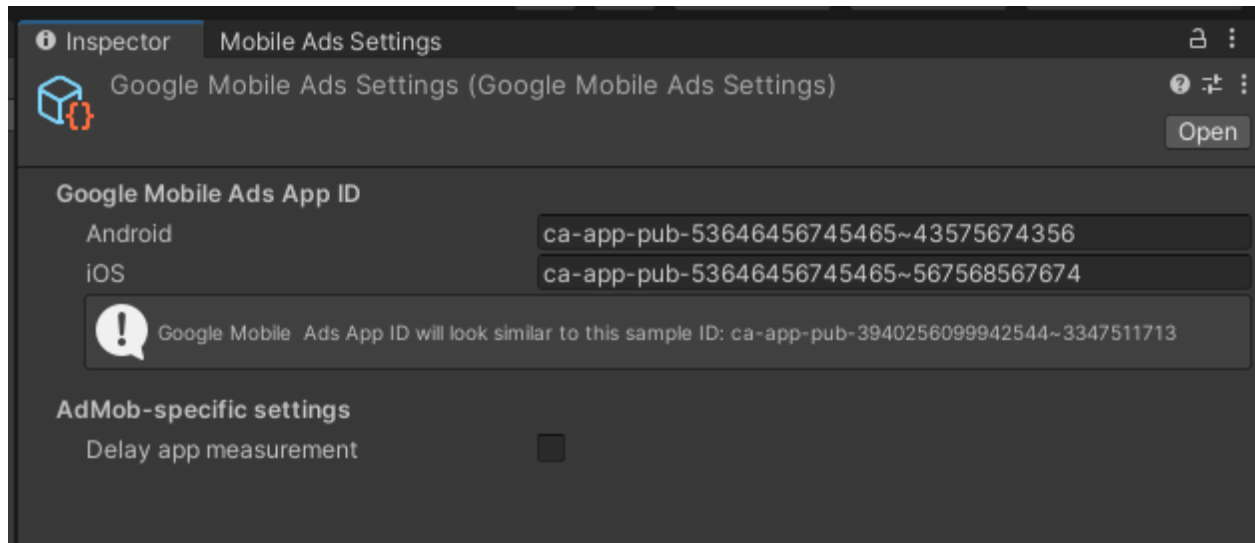
Fill out the AdMob Settings, adding your Ad Unit Ids. You can use the AdMob test ids to test ads first: <https://developers.google.com/admob/android/test-ads>

Step 5

Select **Assets > Google Mobile Ads > Settings** from the menu.



Then in the **Inspector** window, enter your Android and iOS AdMob app ID in each field.



Step 6

Android: Make sure the Play Services Resolver has run by selecting the menu item **Assets > Play Services Resolver > Android Resolver > Resolve**.

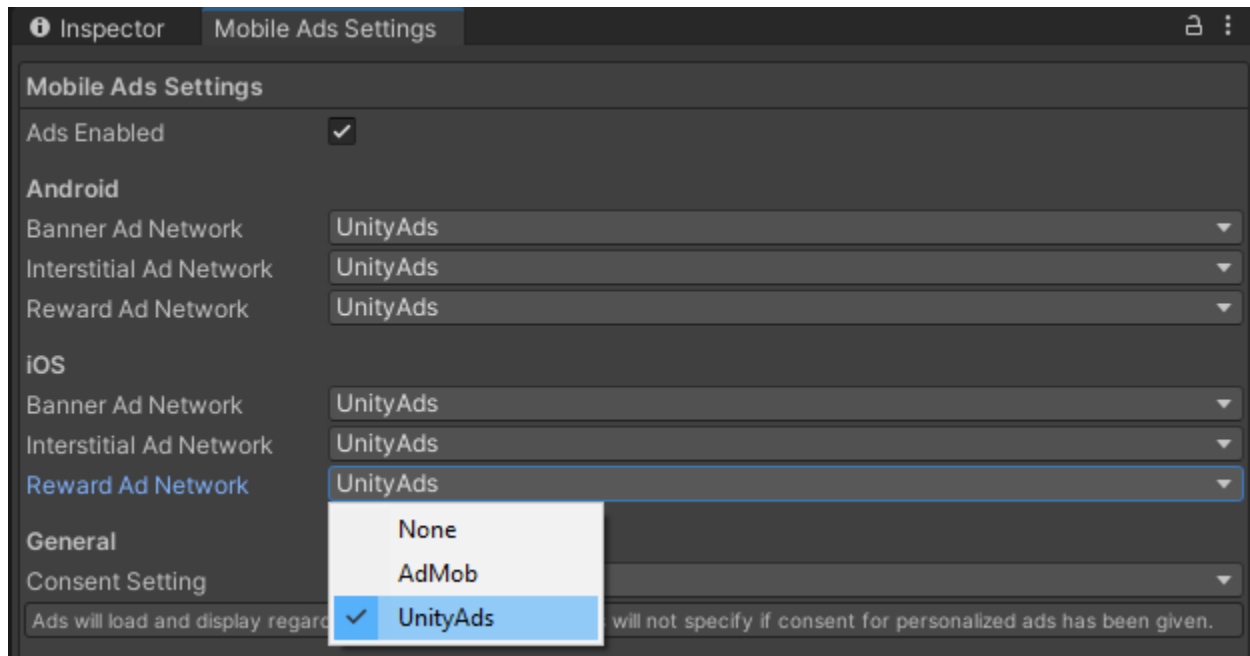
iOS: Make sure Cocoapods has been installed by selecting the menu item **Assets > Play Services Resolver > iOS Resolver > Install Cocoapods**.

You can now test ads by running the game in the editor.

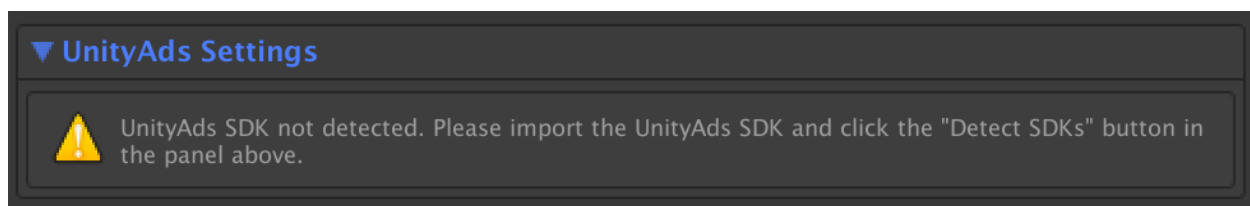
Unity Ads Setup

Step 1

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

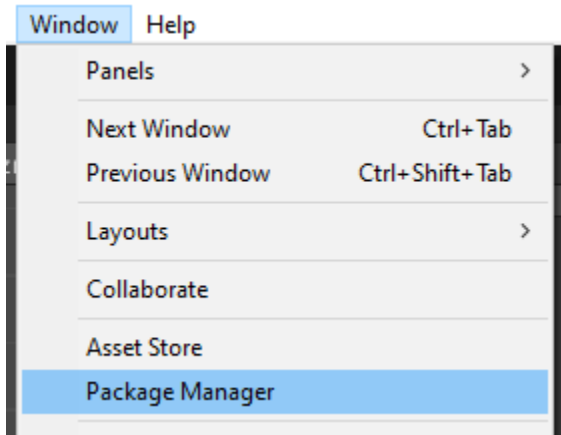


A new section will appear at the bottom of the window called **UnityAds Settings**. Expanding it now will display the following warning:

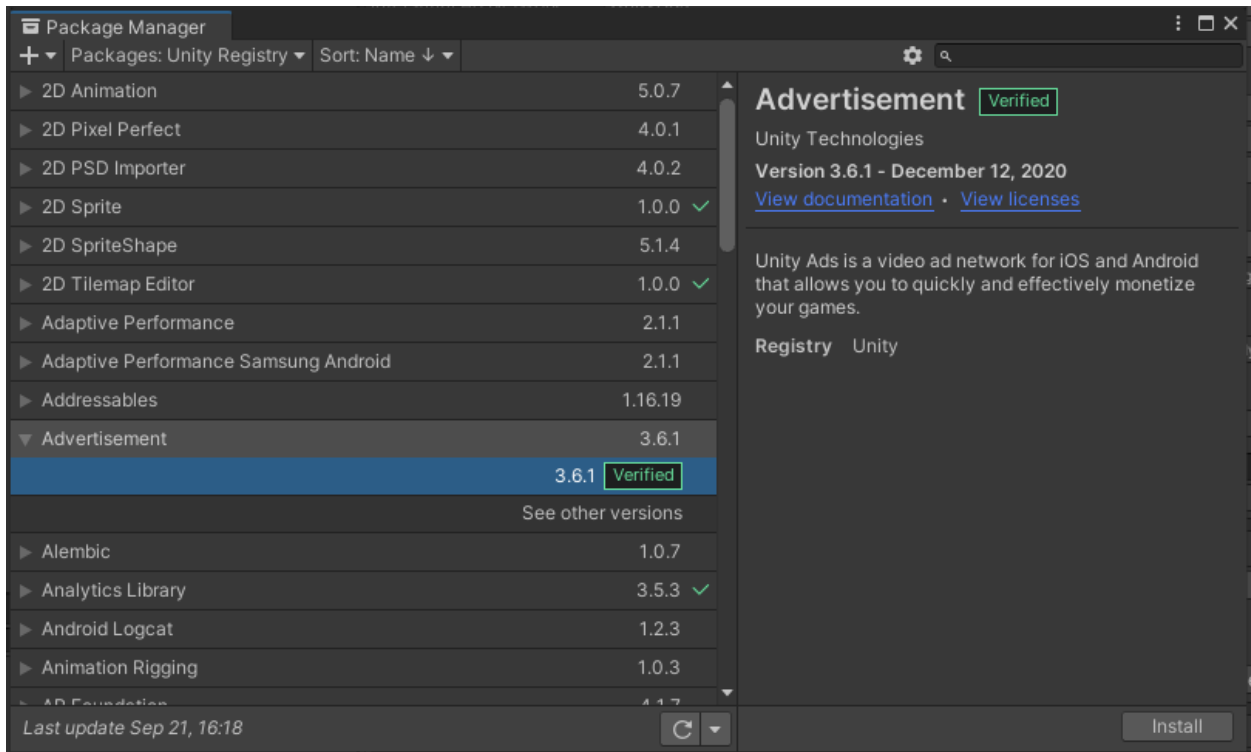


Step 2

Open the **Package Manager** window:

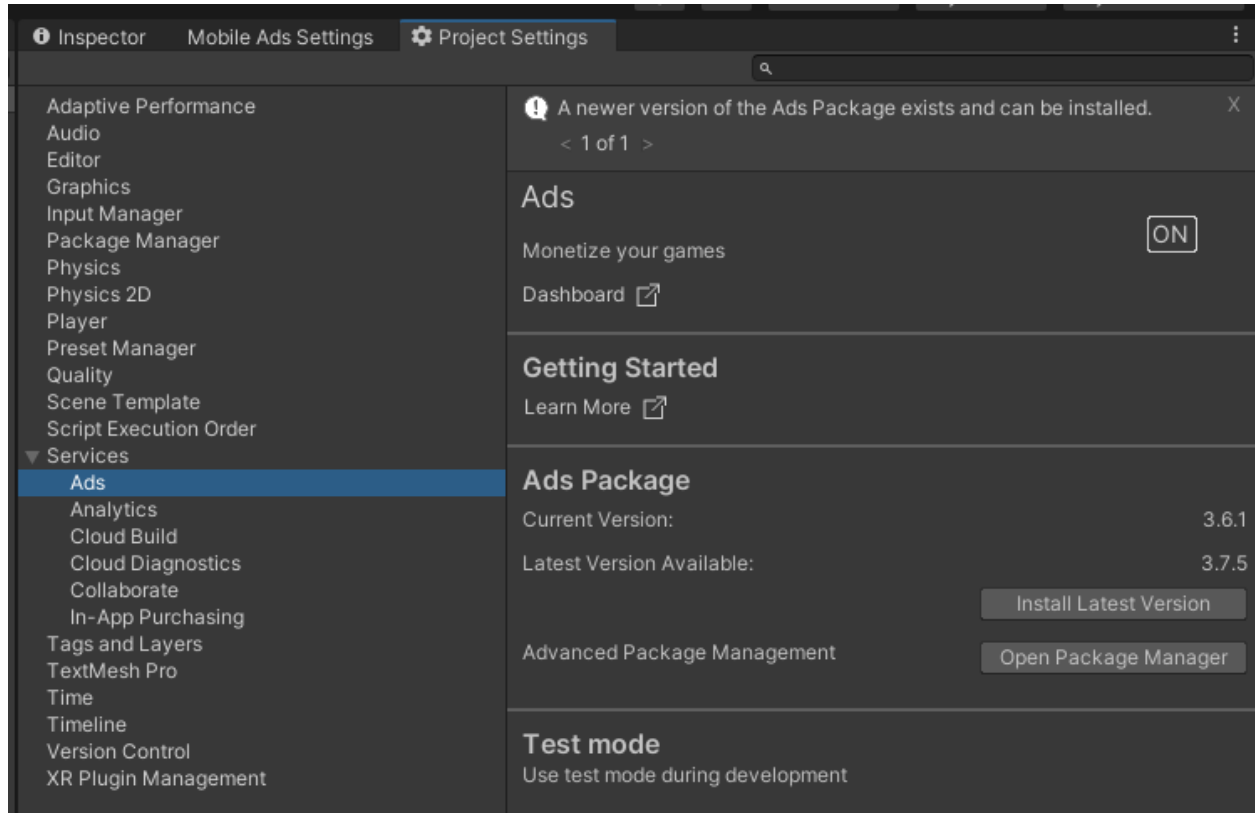


Make sure the **Advertisement** package is installed, if not then install it by clicking the **Install** button located in the bottom right corner:



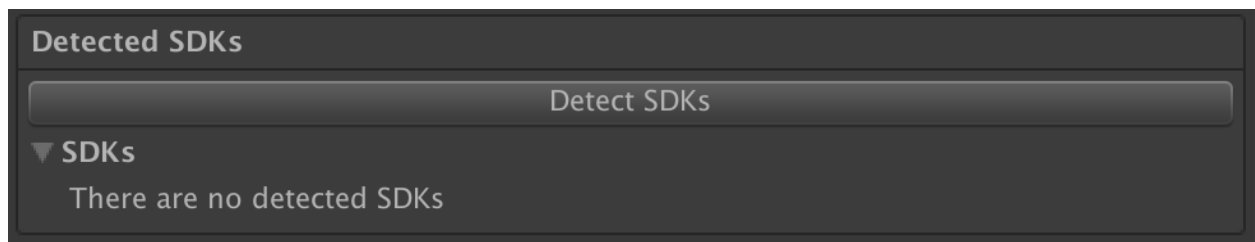
Step 3

Enable Ads in your project by opening the **Project Settings** window (Edit > Project Settings). Then select **Services** then **Ads** on the left and turn Ads on:

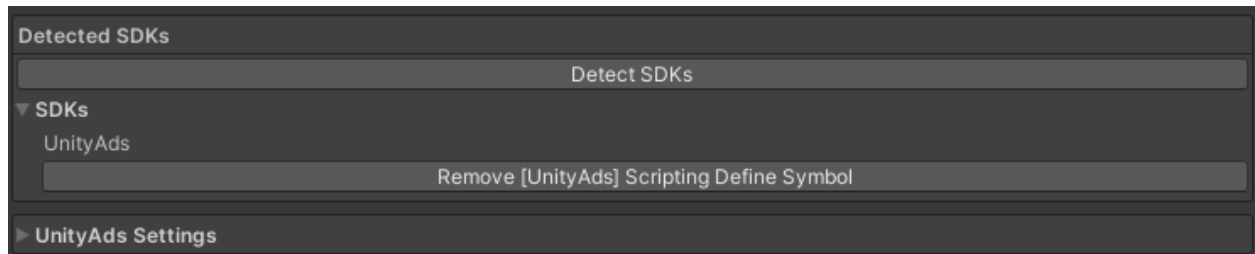


Step 4

Back on the Mobile Ads Settings window, click the **Detect SDKs** button:



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

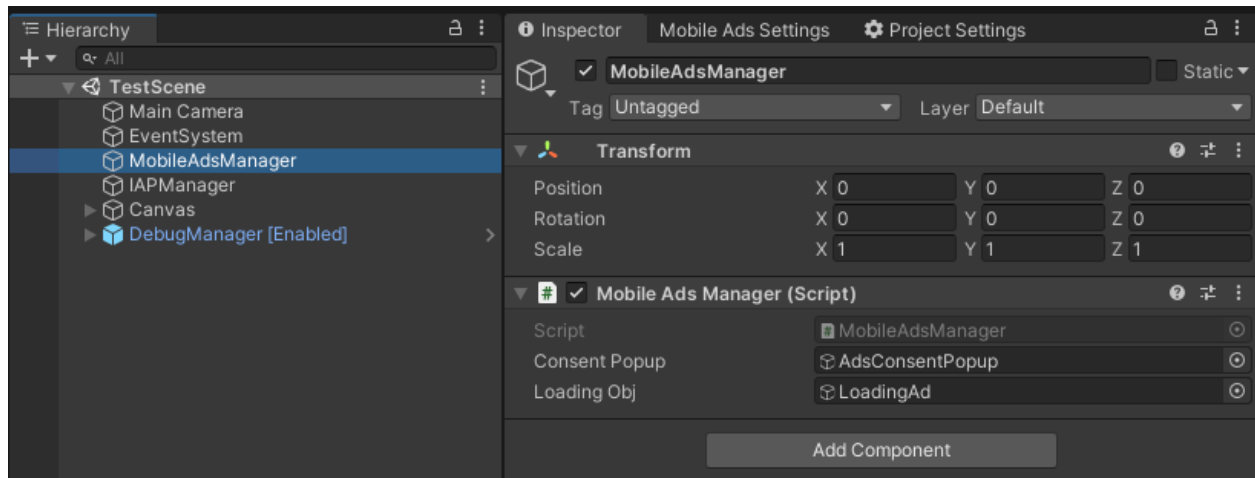


Step 6

Add Your Game Ids and Placement Ids to the fields under UnityAds Settings. You can now test ads by running the game in the editor.

Displaying Ads

To display ads in your game first you need to add the MobileAdsManager component to your scene:



You can then display an ad by call the appropriate method:

```
using BBG.MobileTools;

bool adShown = MobileAdsManager.Instance.ShowInterstitialAd(() => {
    Debug.Log("Ad finished!");
});

if (adShown)
{
    Debug.Log("Ad was displayed to the player!");
}

bool adShown = MobileAdsManager.Instance.ShowRewardAd(
    () => {
        Debug.Log("Ad finished!");
    },
    () => {
        Debug.Log("Reward granted!");
    });

if (adShown)
{
    Debug.Log("Ad was displayed to the player!");
}
```

Test Scene

You easily test ads by opening the included **TestScene** locked in the MobileTools/Scenes folder. Just run the game, click the “Open Debug Console” button, select the **Ads** tab at the top. You can then click one of the buttons to show the ad.