Google Mobile Ad(AdMob)

Anyone can comment

Note: The documentation is deprecated, please use **new web documentation** instead.

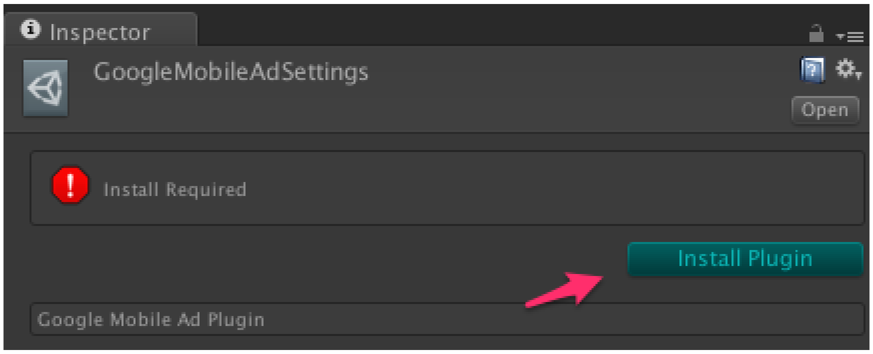
Short Overview

Plugin provides the easy and flexible functionality available in Google Mobile Ad SDK. You will be able to manage and receiving events from banners of all sizes, and interstitial ad.
Plugin Can be used with Android and IOS platforms.

Please read full documentation before using the plugin.
If you're new to IOS app development, please also read IOS Application Setup Guide.
You may want to read about Compilation and signing Android Applications With Unity.

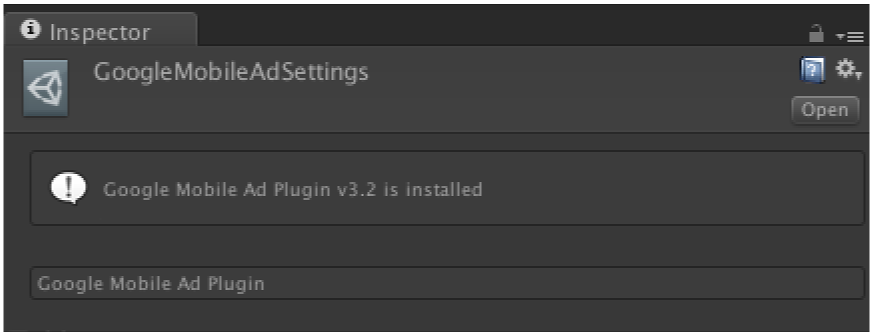
Setup

Plugin may require some small set up action in order to compile in your device with no issue. If plugin isn't installed the settings window will be opened automatically in the Inspector view. It for some reason it wasn't opened after you imported the plugin, go to: **Window → Google Mobile Ad → Edit Settings**



Install Required

And just press **Install Plugin** button. If installation was completed successfully you should see message as on picture below.



Google Mobile Ad Plugin v3.2 is installed

If you have other plugins in your project with requires to override main activity you need to merge such plugins with Google Mobile Ad. You can find necessary info below.

Can I use this plugin with other Android Plugins from Asset Store

How to compile androidnative.jar from eclipse project

How to merge manifest with another android plugin

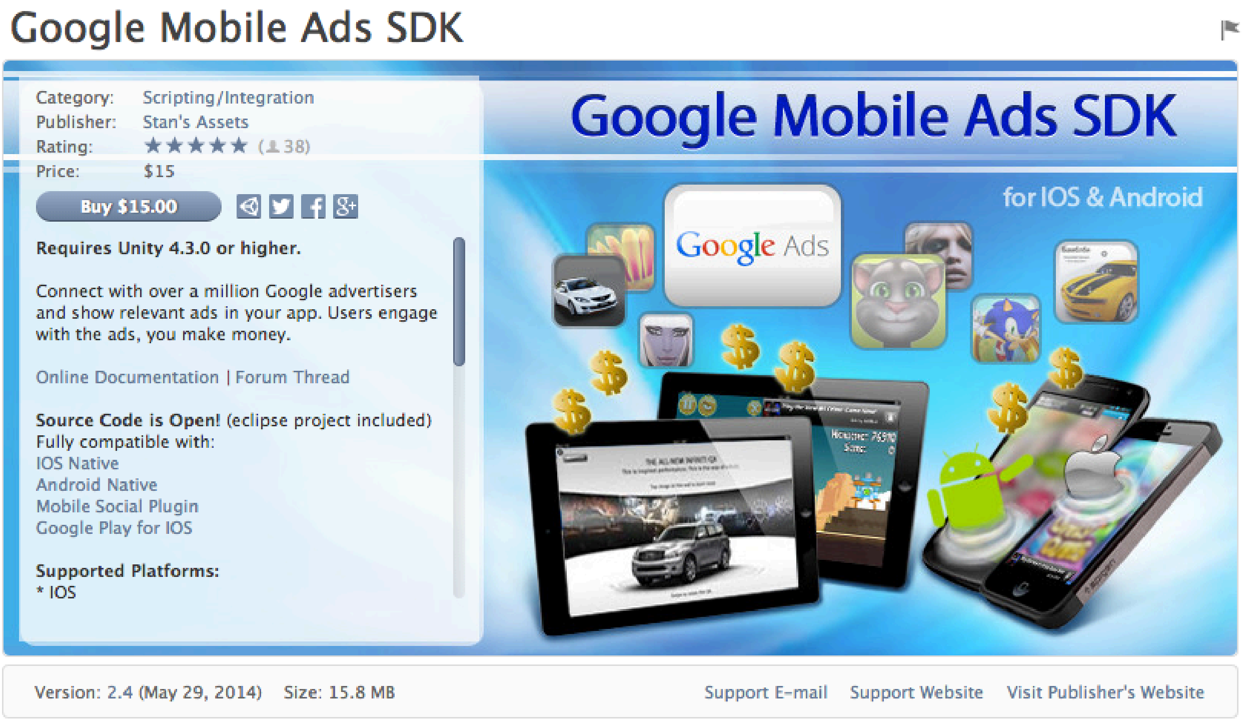
If the plugin has conflict with the IOS / WP8 / Android (with not overrides main activity) Please contact support team.

How to update

1. Version Notes

With every new update I make try to make plugin better. Add new features, improve stability, usability and code base structure.

When new version is available, you can find out what's new in the version and version history by pressing version number on Asset Store Plugin Page:



Google Mobile Ads SDK

Category: Scripting/Integration
Publisher: Stan's Assets
Rating: ★★★★★ (38)
Price: $15

Requires Unity 4.3.0 or higher.

Connect with over a million Google advertisers and show relevant ads in your app. Users engage with the ads, you make money.

Online Documentation | Forum Thread

Source Code is Open! (eclipse project included)
Fully compatible with:
IOS Native
Android Native
Mobile Social Plugin
Google Play for IOS

Supported Platforms:
* IOS

Version: 2.4 (May 29, 2014) Size: 15.8 MB

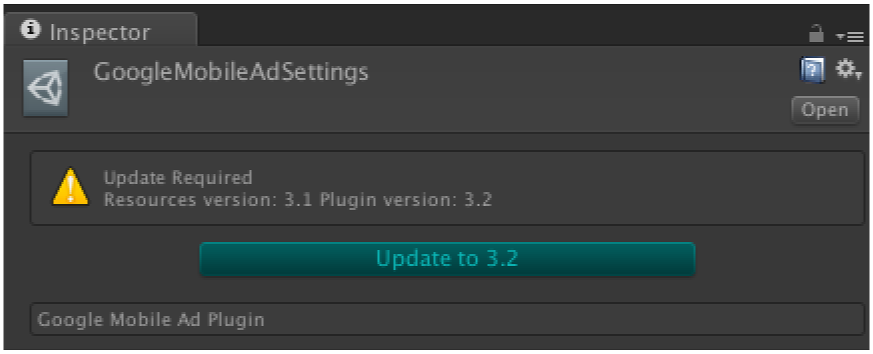
2. Updating

It's recommended to check Version Notes before updating.

Sometimes in order to implement new feature or improve code structure I have to change some of plugin files / folder or method names.

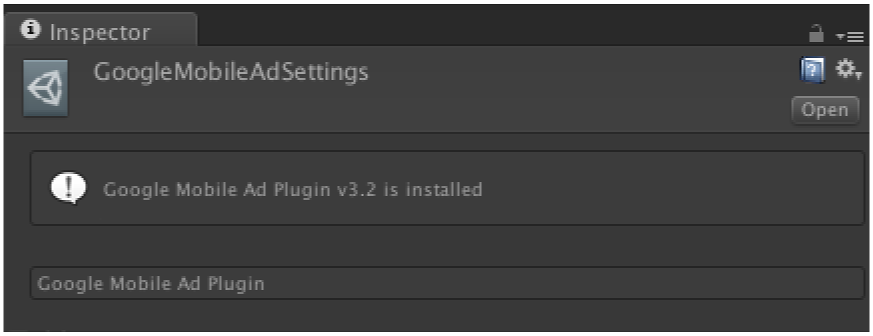
It will be of course described in version notes. But if you simple click update in the Asset Store, you may get duplicated or conflicted files.

After new plugin version is downloaded and unpacked to your project the settings window will be opened automatically in the Inspector view. It for some reason it wasn't opened after you imported the plugin, go to:
Window → Google Mobile Ad→ Edit Settings



Update Required
Resources version: 3.1 Plugin version: 3.2

And just press **Update** button. If installation was completed successfully you should see message as on picture below.



Google Mobile Ad Plugin v3.2 is installed

Released Apps with the plugin

Sky Hoops

Hyper Havoc

POP ISLANDS

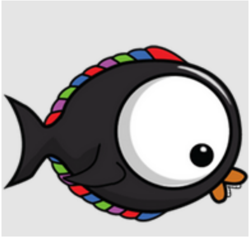
Best Bottle Shoot

High Five HERO!

Battle of Legends

Ninja Karate Defence

Hungry Zombie Saga


Fish Tank

Getting Started

Before you begin

1. Sign up as an AdMob or DFP publisher.
2. Download the SDK for your particular development platform.
3. Familiarize yourself with the AdMob advertising network or DoubleClick For Publishers (DFP) mobile advertising solution.

General Info

The Google Mobile Ads SDK allows developers to easily incorporate mobile-friendly text and image banners as well as rich, full-screen web apps known as interstitials. An ever-growing set of "calls-to-action" are supported in response to user touch including direct access to the App Store, Google Play, Windows Phone 8 Marketplace, iTunes, maps, video and the dialer. Ads can be targeted by location and demographic data. The Google Mobile Ads SDK can be used by the following publisher types:

AdMob publishers

Access the Google AdMob network to easily monetize your application.

DoubleClick For Publishers (DFP) users

Leverage DFP to traffic, target, and serve directly-sold ads. The SDK is available to upgraded DFP (www.google.com/dfp) users for Android and iOS platforms.

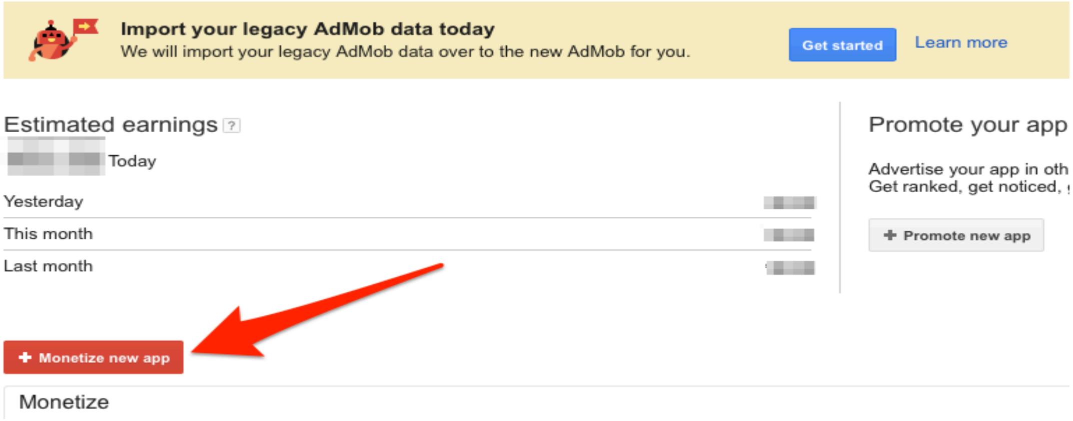
AdSense publishers

Monetize your search results pages with Google search ads.

Setting up google ad with your App

Once you are registered, you can login to the your admob account.

1) Add app for monetization



Import your legacy AdMob data today
We will import your legacy AdMob data over to the new AdMob for you.

Estimated earnings ?

[redacted] Today

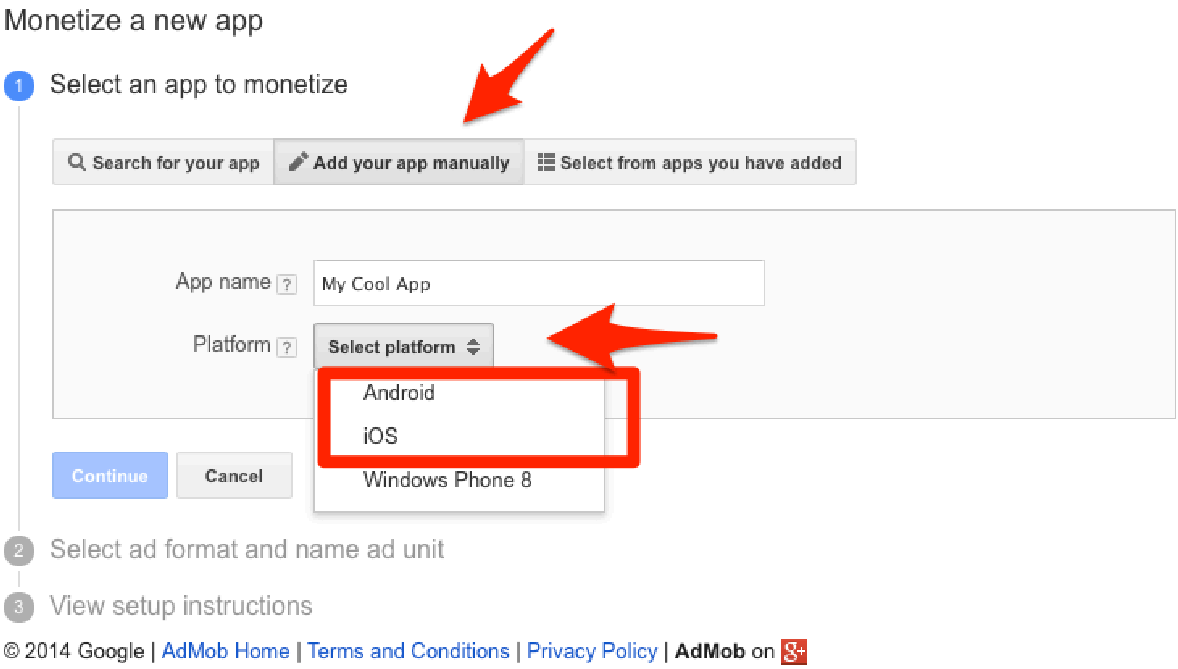
Yesterday	[redacted]
This month	[redacted]
Last month	[redacted]

Promote your app

Advertise your app in oth
Get ranked, get noticed, ,

2) Create the app for monetization:

- Set name for your app (name will not be used in code. so you can specify any name you like).
- Choose Platform, currently plugins support Android and IOS platform.



Monetize a new app

1 Select an app to monetize

App name ? My Cool App

Platform ? Select platform

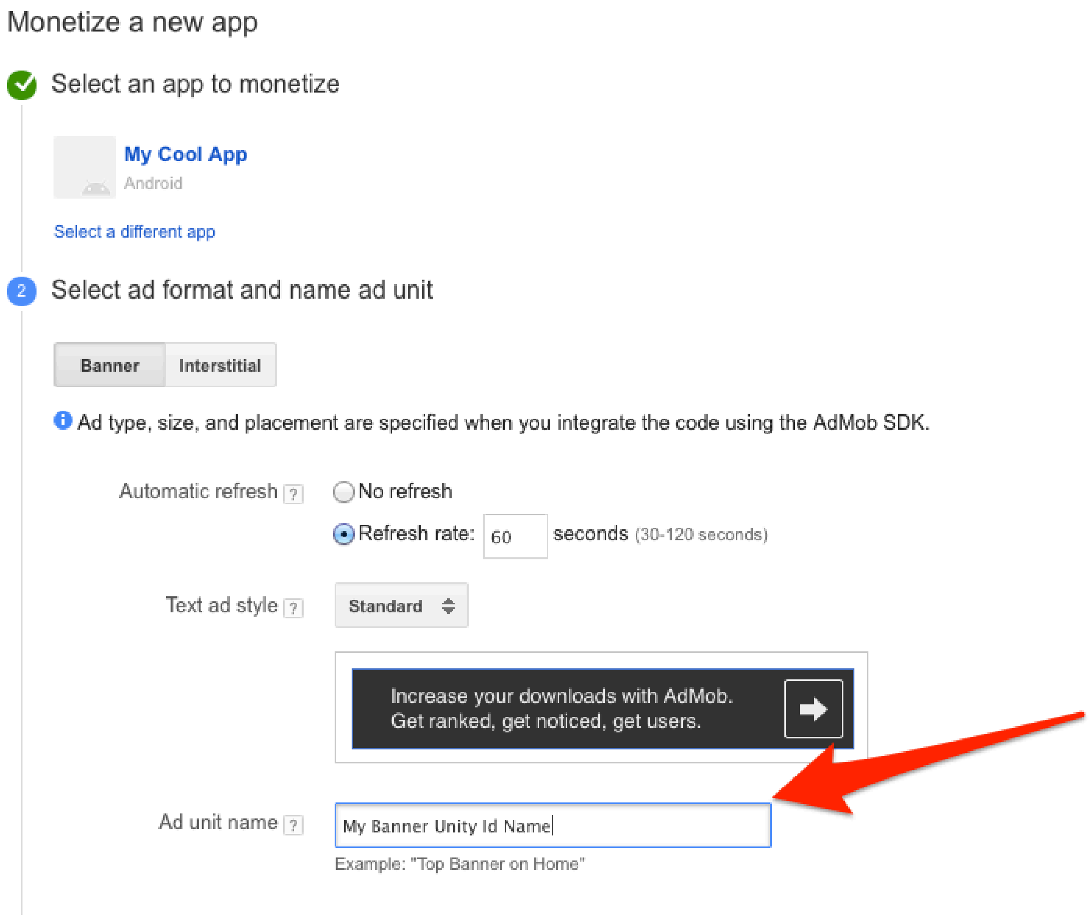
2 Select ad format and name ad unit

3 View setup instructions

© 2014 Google | AdMob Home | Terms and Conditions | Privacy Policy | AdMob on g+

3)Provide the following details for the ad unit:

- **Ad unit name:** Enter a unique name and description that will help you find this ad unit later (e.g., Top Banner on Home).
- **Text ad style:** Select a text ad style that complements your app. You can use the standard style or customize your own style.
- **Automatic refresh** Determine how often a new ad impression is generated. You can choose not to refresh or to refresh ads every 30 to 120 seconds. We recommended refresh rate between 45 and 60 seconds.
- **Google ads:** Select whether or not you'd like to use keyword targeted Google ads and Google certified ad networks to improve your app's fill rate.
- **Click Save.** You'll see the ad unit ID for this ad unit.



Monetize a new app

✓ Select an app to monetize

My Cool App
Android

Select a different app

2 Select ad format and name ad unit

Ad type, size, and placement are specified when you integrate the code using the AdMob SDK.

Automatic refresh ? ○ No refresh
● Refresh rate: 60 seconds (30-120 seconds)

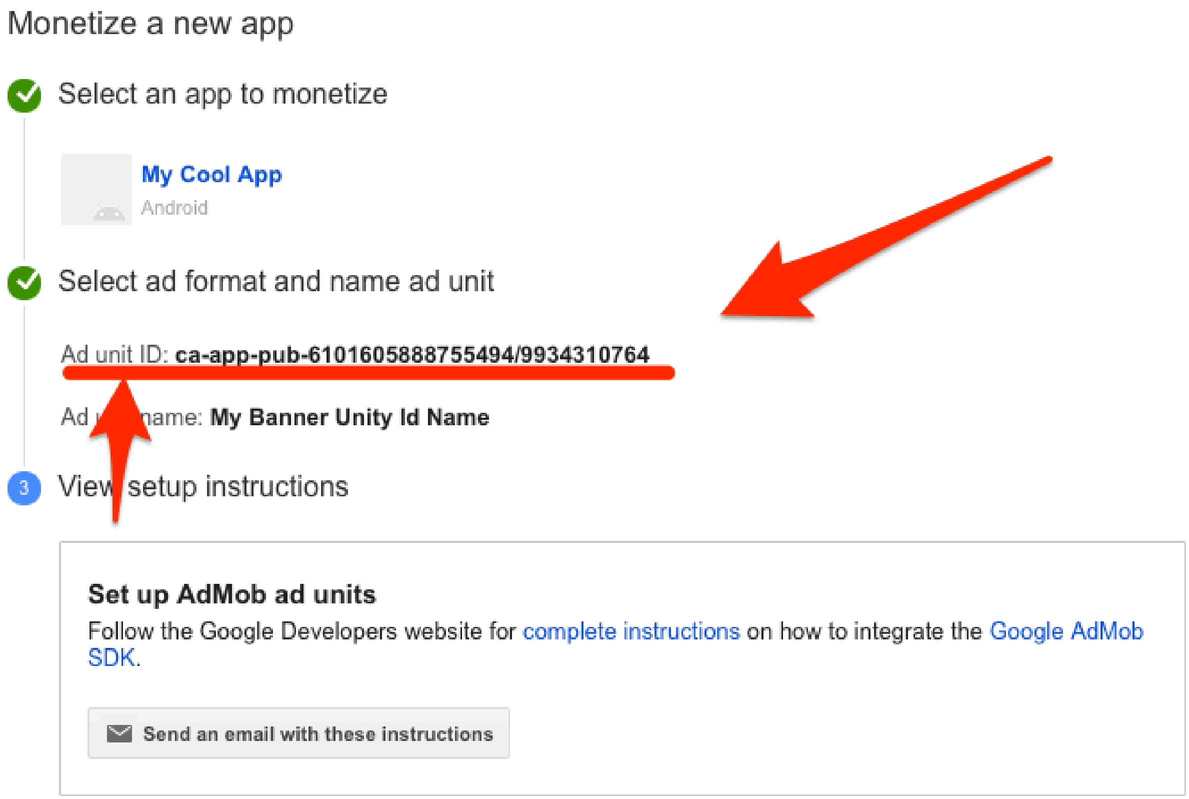
Text ad style ? Standard

Increase your downloads with AdMob.
Get ranked, get noticed, get users.

Ad unit name ? My Banner Unity Id Name

Example: "Top Banner on Home"

4) Copy your **Ad unit id** and click Done. Ad unity Id will be used for displaying the ad in your application.



Monetize a new app

✓ Select an app to monetize

My Cool App
Android

✓ Select ad format and name ad unit

Ad unit ID: **ca-app-pub-6101605888755494/9934310764**

Ad unit name: **My Banner Unity Id Name**

3 View setup instructions

Set up AdMob ad units

Follow the Google Developers website for complete instructions on how to integrate the Google AdMob SDK.

Setup for IOS

Requirements

- iOS version 4.3 or later
- Xcode 4.5 or later

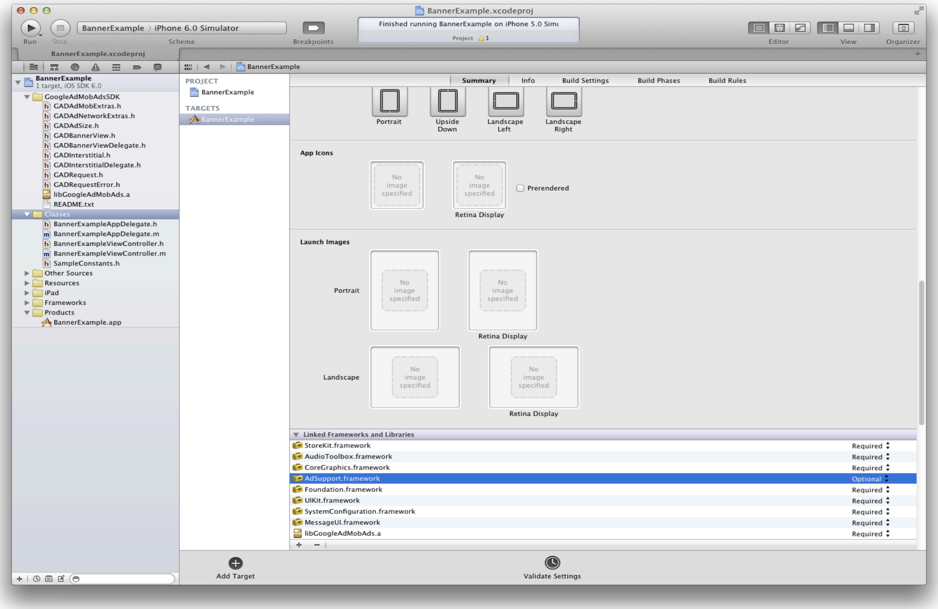
Plugin Setup

Note: This step is automatic after **2.5** version. All frameworks and dependencies will be added automatically on building phase.

1. The SDK library references the following iOS development frameworks which may not already be part of your project:

- CoreTelephony
- MessageUI
- StoreKit
- AdSupport

To add these frameworks, double-click the your project name. O**pen the Link Binary With Libraries** dropdown under the **Build Phases** tab. Add the frameworks from the iOS SDK using the + button that becomes visible.

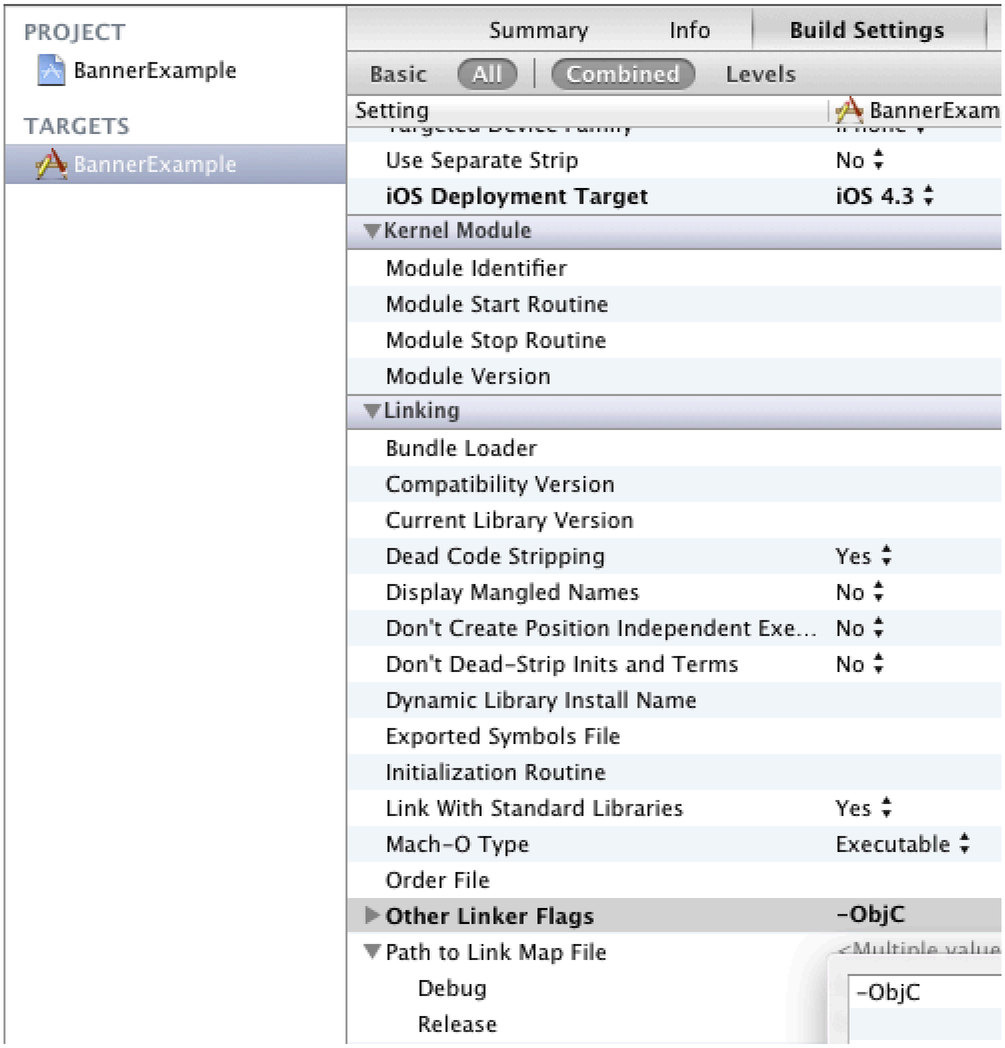


Linked Frameworks and Libraries

StoreKit.framework	Required
AudioToolbox.framework	Required
CoreGraphics.framework	Required
AdSupport.framework	Optional
Foundation.framework	Required
UIKit.framework	Required
SystemConfiguration.framework	Required
MessageUI.framework	Required
libGoogleAdMobAds.a	Required

2. You now need to add -ObjC to the **Other Linker Flags** of your application target's build setting:

1. In Xcode's project navigator, press the blue top-level project icon.
2. Click on your target, then the **Build Settings** tab.
3. Under **Other Linker Flags**, add -ObjC to both **Debug** and **Release**.



Setting	BannerExam
Use Separate Strip	No
iOS Deployment Target	**iOS 4.3**
▼Kernel Module	
Module Identifier	
Module Start Routine	
Module Stop Routine	
Module Version	
▼Linking	
Bundle Loader	
Compatibility Version	
Current Library Version	
Dead Code Stripping	Yes
Display Mangled Names	No
Don't Create Position Independent Exe...	No
Don't Dead-Strip Inits and Terms	No
Dynamic Library Install Name	
Exported Symbols File	
Initialization Routine	
Link With Standard Libraries	Yes
Mach-O Type	Executable
Order File	
▶**Other Linker Flags**	**-ObjC**
▼Path to Link Map File	<Multiple value
Debug	-ObjC
Release	

You should now be able to rebuild your project without any errors. And start using plugin functions.

Setup for Android

Requirements

- *Make sure you have the latest copy of the Android SDK and that you're compiling against at least Android v3.2 (set target in project.properties toandroid-13).*
- *The Google Mobile Ads SDK for Android requires a run-time of Android 2.3 or later (set android:minSdkVersion to at least 9 in yourAndroidManifest.xml). This means you can develop with the latest version of the Android SDK and your app will still run on an earlier Android version (2.3minimum).*

Plugins Setup

Make sure that androidnative.jar and AndroidManifest.xml is inside your **Assets/Plugins/Android** folder.

Open AndroidManifest.xml and enter your package name in the package attribute of the <manifest> element.
package="REPLACE_WITH_YOUR_PACKAGE_ID"   - replace this line with your Package ID

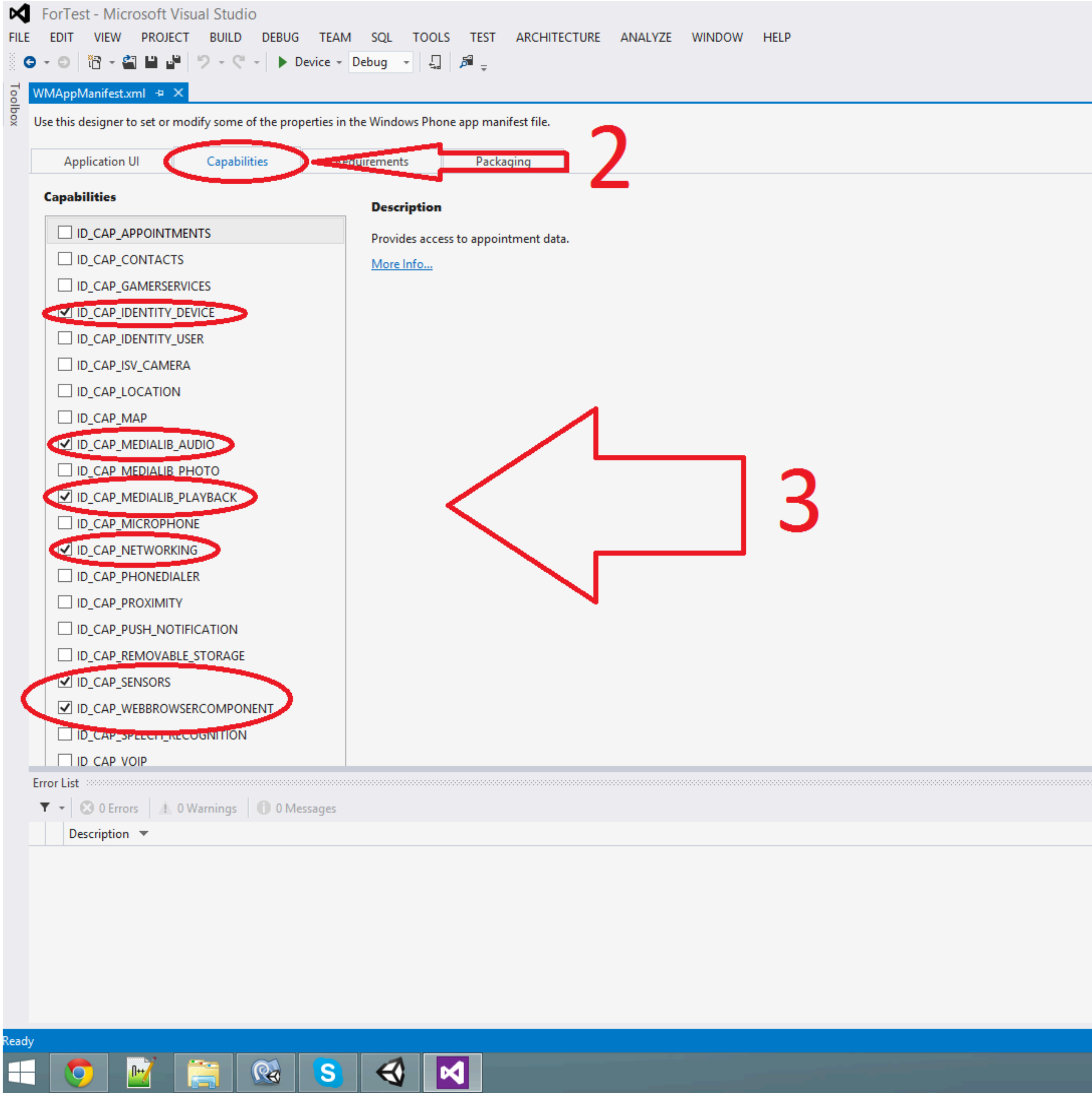
Setup for WP8

Requirements

- *Windows 8 or later*
- *Visual Studio 2012 or later*

Before running the application on a device, make sure that following capabilities are included.

- **ID_CAP_IDENTITY_USER**. (Provides read access used identity)
- **ID_CAP_WEBBROWSERCOMPONENT** (Required since the AdView is a web browser)
- **ID_CAP_IDENTITY_DEVICE**
- **ID_CAP_NETWORKING** (Access to network services is required when requesting ads.)
- **ID_CAP_MEDIALIB_PLAYBACK** (Provides access for currently playing media items)
- **ID_CAP_MEDIALIB_AUDIO** (Provides read access to audio items in media library)



Use this designer to set or modify some of the properties in the Windows Phone app manifest file.

Capabilities

- □ ID_CAP_APPOINTMENTS
- □ ID_CAP_CONTACTS
- □ ID_CAP_GAMERSERVICES
- ☑ ID_CAP_IDENTITY_DEVICE
- □ ID_CAP_IDENTITY_USER
- □ ID_CAP_ISV_CAMERA
- □ ID_CAP_LOCATION
- □ ID_CAP_MAP
- ☑ ID_CAP_MEDIALIB_AUDIO
- □ ID_CAP_MEDIALIB_PHOTO
- ☑ ID_CAP_MEDIALIB_PLAYBACK
- □ ID_CAP_MICROPHONE
- ☑ ID_CAP_NETWORKING
- □ ID_CAP_PHONEDIALER
- □ ID_CAP_PROXIMITY
- □ ID_CAP_PUSH_NOTIFICATION
- □ ID_CAP_REMOVABLE_STORAGE
- ☑ ID_CAP_SENSORS
- ☑ ID_CAP_WEBBROWSERCOMPONENT
- □ ID_CAP_SPEECH_RECOGNITION
- □ ID_CAP_VOIP

Description

Provides access to appointment data.

More Info...

API References

GoogleMobileAd : Singleton<GoogleMobileAd> class.

API methods:

Init ad with your ad unit id. This function should be called before any other function of this class.
public static void Init(string ios_unit_id, string android_unit_id)

Changes ad unity id for banners
public static void SetBannersUnitID(string ad_unit_id)

Changes ad unity id for interstitials
public static void SetInterstisialsUnitID(string ad_unit_id)

Creates banner ad using TextAnchor

public static GoogleMobileAdBanner CreateAdBanner(TextAnchor anchor, GADBannerSize size)

Creates banner ad using custom x/y position
public static GoogleMobileAdBanner CreateAdBanner(int x, int y, GADBannerSize size)

Destroy banner by id
public static void DestroyBanner(int id)

Add keyword for targeting purposes
public static void AddKeyword(string keyword)

Sets the user's birthday for targeting purposes.
public static void SetBirthday(int year, AndroidMonth month, int day);

Set gender for targeting purposes, use GADGenger
public static void SetGender(GoogleGenger gender)

This method allows you to specify whether you would like your app to be treated as child-directed for purposes of the Children’s Online Privacy Protection Act (COPPA) - http://business.ftc.gov/privacy-and-security/childrens-privacy.

If you set this method to true, you will indicate that your app should be treated as child-directed for purposes of the Children’s Online Privacy Protection Act (COPPA).

If you set this method to false, you will indicate that your app should not be treated as child-directed for purposes of the Children’s Online Privacy Protection Act (COPPA).

If you do not set this method, ad requests will include no indication of how you would like your app treated with respect to COPPA.

By setting this method, you certify that this notification is accurate and you are authorized to act on behalf of the owner of the app. You understand that abuse of this setting may result in termination of your Google account.

Note: it may take some time for this designation to be fully implemented in applicable Google services.

```
public static    void TagForChildDirectedTreatment(bool val);
```

Causes a device to receive test ads. The deviceId can be obtained by viewing the logcat output after creating a new ad.

The device ID that AdMob accepts is a hashed value (I'm not sure, but it might also include a salt) of your actual device ID. The way to get this hashed device ID is to make a live request on your device and check logcat for a message like "To get ads no this device, call adRequest.addTestDevice("YOUR_HASHED_DEVICE_ID")". This ID is what you should use. It should be a 32-digit HEX number like the numbers the OP has.

```
public static void AddTestDevice(string deviceId)
public static void AddTestDevices(params string[] ids)
```

Function will start interstitials banner request and will show it as soon as banner loaded.

```
public static void StartInterstitialAd()
```

```
Function will send interstitials banner request.
public static void LoadInterstitialAd()
```

```
Shows interstitial banner if it was previously loaded
public static void ShowInterstitialAd()
```

```
Record IAP resolutions. Read More.
public static void RecordInAppResolution(GADInAppResolution resolution)
```

Called when interstitial an ad is received
ON_INTERSTITIAL_AD_LOADED

Called when interstitial an ad request failed
ON_INTERSTITIAL_AD_FAILED_LOADING

Called when interstitial an ad opens an overlay that covers the screen.
ON_INTERSTITIAL_AD_OPENED

Called when the user is about to return to the application after clicking on an ad.
ON_INTERSTITIAL_AD_CLOSED

Called when an ad interstitial leaves the application (e.g., to go to the browser).
ON_INTERSTITIAL_AD_LEFT_APPLICATION

Called when ad action trigers in-app request. Read More
ON_AD_IN_APP_REQUEST

Warning: GoogleMobileAd not event dispather by it self. To be able to listen for the events sign on public static GoogleMobileAdInterface controller getter events of GoogleMobileAd class. Controller will be created after GoogleMobileAd **init** function.

GoogleMobileAdBanner interface.

API methods:

Hide ad banner

```
public void HideAd()
```

Show ad banner (only if it was hidden by **HideAd** *function)*

```
public void ShowAd()
```

Refresh ad content (will send new request to google)

```
public void Refresh()
```

Change Banner Position without destroying the banner

```
public void SetBannerPosition(int x, int y)
public void SetBannerPosition(TextAnchor anchor)
```

Get / Set:

Banner id

```
int id {get;}
```

Banner width

```
int width {get;}
```

Banner height

```
int height {get;}
```

true if banner was loaded

```
bool IsLoaded {get;}
```

true if banner currently on screen

```
bool IsOnScreen {get;}
```

Defines show or not banner when it’s loaded.

```
bool ShowOnLoad{get; set;}
```

Events:

Called when an ad is received

ON_BANNER_AD_LOADED

Called when an ad request failed
ON_BANNER_AD_FAILED_LOADING

Called when an ad opens an overlay that covers the screen.
ON_BANNER_AD_OPENED

Called when the user is about to return to the application after clicking on an ad.
ON_BANNER_AD_CLOSED

Called when an ad leaves the application (e.g., to go to the browser).
ON_BANNER_AD_LEFT_APPLICATION

InAppPurchaseListener

Note: You will only receive in-app purchase (IAP) ads if you specifically configure an IAP ad campaign in the AdMob front end.

Implement the onInAppPurchase listener, is really easy, all you have to do is to subscribe to **ON_AD_IN_APP_REQUEST** event

```
GoogleMobileAd.addEventListener(GoogleMobileAdEvents.ON_AD_IN_APP_REQUEST, OnInAppRequest);
```

Implement Event data will contain product id. You shoud start your game purchase flow with this id as soon as you will receive **ON_AD_IN_APP_REQUEST** event

```
private void OnInAppRequest(CEvent e) {
        //getting product id
        string productId = (string) e.data;
        Debug.Log ("In App Request for product Id: " + productId + " received");
        //Start purchase flow with productId here
}
```

Once the purchase is complete, you should call `RecordInAppResolution` with one of the following constants defined in GADInAppResolution:

```
public enum GADInAppResolution  {
    RESOLUTION_SUCCESS = 0,        // Purchase was a success
    RESOLUTION_FAILURE = 1,         // Error while processing purchase
    RESOLUTION_INVALID_PRODUCT = 2, // Error while looking up product
    RESOLUTION_CANCELLED = 3       // Purchase was cancelled by user
}
```

An example of a success call would look like this:

```
GoogleMobileAd.RecordInAppResolution(GADInAppResolution.RESOLUTION_SUCCESS);
```

PlayMaker Actions

The plugin now contains playmaker actions.

The actions scripts can be found in the zip archive at:

Assets/Extensions/GoogleMobileAd/Addons/PlayMakerActions

You can simply unrar it to the same folder and Google Mobile Ad actions will appear under playmaker actions menu. You always welcome on the PlayMaker Actions Forum Thread to request new actions or report a bug

The current action list is:

- GAD_InitGoogleAd
- GAD_BannerEvents
- GAD_InterstitialEvents
- GAD_SetAdTargeting
- GAD_SetAdTestDevices
- GAD_CreateBanner
- GAD_ShowBanner
- GAD_HideBanner
- GAD_RefreshBanner
- GAD_DestroyBanner
- GAD_StartInterstisialAd
- GAD_LoadInterstisialAd
- GAD_ShowInterstisialAd

Implementation Guide

Make sure you read the Google Mobile Ad Guide section first.

With iAd you can use two types of ad interstitial and banners. But first of all you need to init google ad using the **GAD_InitGoogleAd** action.

1. Interstitial

There is two ways to use interstitial.

- **GAD_StartInterstisialAd** action. Will start Ad load request and as soon the ad loaded will show the interstitial ad.

- Use **GAD_LoadInterstisialAd** to load the Ad content. **SuccessEvent** will be fired if content was loaded successfully and **failEvent** if not. After content was successfully loaded you may use **GAD_ShowInterstisialAd** action to show the ad.

You should use one of those approaches every time you want to show the interstitial ad.

2. Banners

1) Create ad banner using **GAD_CreateBanner** action. If banner is successfully created **SuccessEvent** will be fired and **FailEvent** if creation failed.

Do not fill the bannerId section. It will be filled after banner is create. Store this id. It will be used to control create banner with following playmaker actions

- **GAD_HideBanner** - will hide the banner
- **GAD_ShowBanner** - will show the banner
- **GAD_RefreshBanner** - will refresh the banner
- **GAD_DestroyBanner** - will destroy the banner

Frequently Asked Questions

I have one unit id for banners ad and one for interstitial, but init function is taking only one id. How should I specify both of them?

If you have two ids to specify, you can init admob controller with for example banners id, and then set id specifically for interstitial

How do I get an AdMob ad unit ID?

Directions for how to create an AdMob ad unit ID can be found here. AdMob ad unit IDs have the form ca-app-pub-XXXXXXXXXXXXXXXX/NNNNNNNNNN.

I keep getting the error 'The Google Play services resources were not found. Check your project configuration to ensure that the resources are included.

You can safely ignore this message. Your app will still fetch and serve banner ads.

I keep getting the error 'Invalid unknown request error: Cannot determine request type. Is your ad unit id correct?'

Make sure your ad unit ID is correct. For publishers using the new AdMob front end, the ad unit ID will be in the form ca-app-pub-XXXXXXXXXXXXXXXX/NNNNNNNNNN. You will get this error if you use the form pub-XXXXXXXXXXXXXXXX.

My app support autorotation. But ad banner is not changing when the app is rotated.

The auto rotation for banners is not supported. But you can implement it by your self, I will describe algorithm below.
Most of game apps have one orientation, and those who support both as usual us

different banner position and size for different orientation. that why I do not see the reason to implement automation auto rotation.

1) 95% if user will not be use this
2) Users who will need this feature would love to add extra enchantments

So I decided to give you full control on banners instead of implementing features that you will not use.

Here is algorithm how you can use to implement custom auto rotate banner to your app.

1) App started at landscape.
2) Create banner and assign it to **LandscapeBanner** variable
3) Detected rotation to portrait
4) hide **LandscapeBanner**
5) Create new banner and assign it to **PortraitBanner** variable
6) Detected rotation to back to landscape
7) check if **LandscapeBanner** was created, if no see the step 2.
8) Hide **PortraitBanner.** Show **LandscapeBanner**

Example Scenes

Example

This example scene can be found at **Assets/Extensions/GoogleMobileAd/Example/Scenes/Example.**

Scene will demonstrate usage of Google Mobile Ad API.

The controller script **GoogleAdsExample.cs** is attached to the **GoogleAdUseExample** gameobject and provides example for API calls.

[**Setup**]

Scene will work out of the box after plugin installation. However example scene will use unity_id's from my test account.

Get your own ids after google ad account set up. Open the **Window-> GoogleMobileAds -> Edit Settings** and enter id's in following lines.

```
//replace with your ids
BannerAdUnitAd ="ca-app-pub-6101605888755494/1824764765";
InterstitialsAdUnitAd= "ca-app-pub-6101605888755494/3301497967";
```

NoCodingExample

This example scene can be found at **Assets/Extensions/GoogleMobileAd/Example/Scenes/NoCodingExample.**

Scene will demonstrate usage of Google Mobile Ad API without any actions.

The controller scripts for Banner **AdMobBanner.cs** and for Interstitial **AdMobBannerInterstitial.cs** are attached to the **MyBanner/MyInterstitial** gameobjects and provides examples for API calls.

[**Setup**]

Scene will work out of the box after plugin installation. However example scene will use unity_id's from my test account.