



Android Unity Plugin v4.0
Publishers



Android Unity Plugin v4.0

Publishers



Introduction

This document outlines the Android Unity Plugin v4.0 that allows displaying mobile ads, Editorials and interstitials (**simple** and **HTML5** version) inside android applications developed by **Unity**.

The current version of the Android SDK is compatible with Android 2.3 and above.

Today users expect the same high-quality experience regardless of device and if they don't get it, they blame your brand. This is why we use **Akamai number 1 CDN world wide** to display your ads.

The new Android SDK v4.0 for Unity supports now **Google's Advertising Identifier**.

The new Android SDK v4.0 for Unity allows you handling ads events.

1. Download Android SDK

Download the Ad4game Android SDK publishers for Unity. Decompress the zip file and extract the files to your development computer. The Android SDK for Unity is provided as three java JAR files (**AndroidSDKPublisher4.0.jar**, **a4gpublisherunity.jar** et **google-play-services.jar**), making it easy to include in your Android project.

2. Setup the Unity Project

Create a 'Plugins' folder and within this create another folder called 'Android' (**Assets->Plugins->Android**). Copy the **AndroidSDKPublisher4.0.jar**, **a4gpublisherunity.jar** and the **google-play-services.jar** into this folder.

- Create a 'res' folder and within this create another folder called 'values' (**Assets->Plugins->Android->res->values**). Copy the **version.xml** to this folder.

Build the project for Android and then go into the 'Temp\StagingArea' folder with your project folder and copy the **AndroidManifest.xml** file into the 'Android' folder. This manifest file now needs to be edited to look like this:



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:installLocation="preferExternal"
android:theme="@android:style/Theme.NoTitleBar" package="com.a4gpublisherunity" android:versionName="1.0"
android:versionCode="1">
    <supports-screens android:smallScreens="true" android:normalScreens="true" android:largeScreens="true"
android:xlargeScreens="true" android:anyDensity="true" />
    <application android:icon="@drawable/app_icon" android:label="@string/app_name" android:debuggable="false">
        <activity android:name=".A4GPublisherUnity" android:launchMode="singleTask"
android:label="@string/app_name" android:configChanges="fontScale|keyboard|keyboardHidden|locale|mnc|mcc|
navigation|orientation|screenLayout|screenSize|smallestScreenSize|uiMode|touchscreen"
android:screenOrientation="sensor">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name="com.unity3d.player.UnityPlayerActivity" android:launchMode="singleTask"
android:label="@string/app_name" android:configChanges="fontScale|keyboard|keyboardHidden|locale|mnc|mcc|
navigation|orientation|screenLayout|screenSize|smallestScreenSize|uiMode|touchscreen"
android:screenOrientation="sensor">
        </activity>
        <activity android:name="com.unity3d.player.UnityPlayerNativeActivity" android:launchMode="singleTask"
android:label="@string/app_name" android:configChanges="fontScale|keyboard|keyboardHidden|locale|mnc|mcc|
navigation|orientation|screenLayout|screenSize|smallestScreenSize|uiMode|touchscreen"
android:screenOrientation="sensor">
            <meta-data android:name="android.app.lib_name" android:value="unity" />
            <meta-data android:name="unityplayer.ForwardNativeEventsToDalvik" android:value="false" />
        </activity>
    </application>
    <uses-feature android:glEsVersion="0x00020000" />
    <uses-sdk android:minSdkVersion="6" android:targetSdkVersion="17" />
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.GET_TASKS" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
</manifest>
```

3. Display Mobile ads

You will create a C# file that will attach to a Unity GameObject as a script. The C# file need to be under the **Assets** folder.

You'll then need to call the **loadTopAds** or **loadBottomAds** (you can call the both methods to display two ads (one on the top, and the other on the bottom of the screen) methods in the **Awake()** function of an empty GameObject in your startup scene:

```
AndroidJavaClass jc = new AndroidJavaClass("com.unity3d.player.UnityPlayer");
AndroidJavaObject a4g = jc.GetStatic<AndroidJavaObject>("currentActivity");
a4g.Call("loadTopAds", "yourZoneID");
a4g.Call("loadBottomAds", "yourSecondZoneID");
```

3.1 Handling Events

If you want to handle ads events, you need first to set your GameObject in the **Awake()** function as follow :

```
a4g.Call("setGameObject", "game object name");// Main Camera for example
```

Then, you can call the following functions :

```
void onAdLoaded() {
    // Do your treatment
}
void onAdFailed() {
    // Do your treatment
}
void onAdClicked() {
    // Do your treatment
}
```

Now, go to **File ->Build Settings ->Player Settings**, and change the Bundle Identifier to **com.a4gpublisherunity**.

Build the Unity project for Android platform and run it on your device to see the ads.

4. Display Android Editorials

To display Android Editorials inside your unity project, you need to use one of the three methods (you can use all of them) bellow:

- 1- **showA4GEditos** : If you want to display the editorials once the activity's started.
- 2- **showA4GEditosWithStartTime** : If you want to display the editorials after certain time.
- 3- **showA4GEditosWithStartAndEndTime** : If you want to display the editorials after certain time and close them automatically after certain time.
- 4- **showA4GEditosOnClick** : If you want to display the editorials when the user click on a button.

Note: The three first methods mentioned above are valid only if your project target android **3.0** or above, **showA4GEditoOnClick** is valid from android **2.3** or above.

```
using UnityEngine;
using System.Collections;

public class Editos : MonoBehaviour {

    AndroidJavaClass jc;
    AndroidJavaObject a4g;

    void Awake() {
        jc = new AndroidJavaClass("com.unity3d.player.UnityPlayer");
        a4g = jc.GetStatic<AndroidJavaObject>("currentActivity");

        //Load Top Ads with zoneid = 16209
        a4g.Call("loadTopAds", "16209");

        // Load Bottom Ads with zoneid = 33055
        a4g.Call("loadBottomAds", "33055");

        // Display Editorials once the activity's started
        a4g.Call("showA4GEditos", "32593");

        // Display Editorials after 7 seconds
        a4g.Call("showA4GEditosWithStartTime", "32593", 7000);

        // Display Editorials after 15 seconds and close them after 20 seconds
        a4g.Call("showA4GEditosWithStartAndEndTime", "32593", 15000, 20000);
    }

    void OnGUI () {
        // Create a button to Display Editorials on the click

        if(GUI.Button(new Rect(20,300,80,40), "Editorials")) {
            a4g.Call("showA4GEditosOnClick", "32593");
        }
    }
}
```

4.1 Handling Events

If you want to handle Editorials events, you need first to set your GameObject in the **Awake()** function as follow :

```
a4g.Call("setGameObject", "game object name");// Main Camera for example
```

Then, you can call the following functions :

```
void onEditorialLoaded() {  
    // Do your treatment  
}  
void onEditorialFailed() {  
    // Do your treatment  
}  
void onEditorialClicked() {  
    // Do your treatment  
}  
void onEditorialClosed() {  
    // Do your treatment  
}
```

5. Display Android Interstitials

To display Android interstitials inside your unity project, you need to use one of the three methods (you can use all of them) bellow:

- 1- **showA4GInterstitials** : If you want to display the interstitials once the activity's started.
- 2- **showA4GInterstitialsWithStartTime** : If you want to display the interstitials after certain time.
- 3- **showA4GInterstitialsWithStartAndEndTime** : If you want to display the interstitials after certain time and close them automatically after certain time.

```
using UnityEngine;
using System.Collections;

public class CallA4GPublisher : MonoBehaviour {

    AndroidJavaClass jc;
    AndroidJavaObject jo;

    void Awake() {
        jc = new AndroidJavaClass("com.unity3d.player.UnityPlayer");
        jo = jc.GetStatic<AndroidJavaObject>("currentActivity");

        //Display interstitials when the activity's started
        jo.Call("showA4GInterstitials", "24992");

        //Display interstitials after 5 seconds
        jo.Call("showA4GInterstitialsWithStartTime", "24992", 5000);

        //Display interstitials after 10 seconds and close them after 20 seconds
        jo.Call("showA4GInterstitialsWithStartAndEndTime", "24992", 10000, 20000);
    }
}
```

5.1 Handling Events

If you want to handle Interstitials events, you need first to set your GameObject in the **Awake()** function as follow :

```
jo.Call("setGameObject", "game object name");// Main Camera for example
```

Then, you can call the following functions :

```
void onInterstitialsLoaded() {
    // Do your treatment
}

void onInterstitialsFailed() {
    // Do your treatment
}

void onInterstitialsClicked() {
    // Do your treatment
}

void onInterstitialsClosed() {
    // Do your treatment
}
```