



Android Unity Plugin v2.0
Advertisers



ad4game
GAME ADVERTISING NETWORK
MOBILE VERSION (ANDROID) 

Android Unity Plugin v2.0

Advertisers



Introduction

This document outlines the Android Unity Plugin v2.0 that allows tracking installations of android applications developed by **Unity**.

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

The new Android SDK v2.0 for Unity supports now **Google's Advertising Identifier**.

1. Download Android SDK

Download the Ad4game Android SDK advertisers 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 (**AndroidSDKTracker2.0.jar**, **A4GTrackerUnity.jar** et **google-play-services.jar**), making it easy to include in your Unity project.

2. Setup the Unity Project

- Create a 'Plugins' folder and within this create another folder called 'Android' ([Assets->Plugins->Android](#)). Copy the **AndroidSDKTracker2.0.jar**, **A4GTrackerUnity.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.a4gtrackerunity" 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=".A4GTrackingUnity" 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>

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



You will then 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 **loadTrack** method 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("loadTrack", "yourTrackerID");
```

Now, go to **File ->Build Settings ->Player Settings**, and change the Bundle Identifier to **com.a4gtrackerunity**.