



Addience SDK Installation

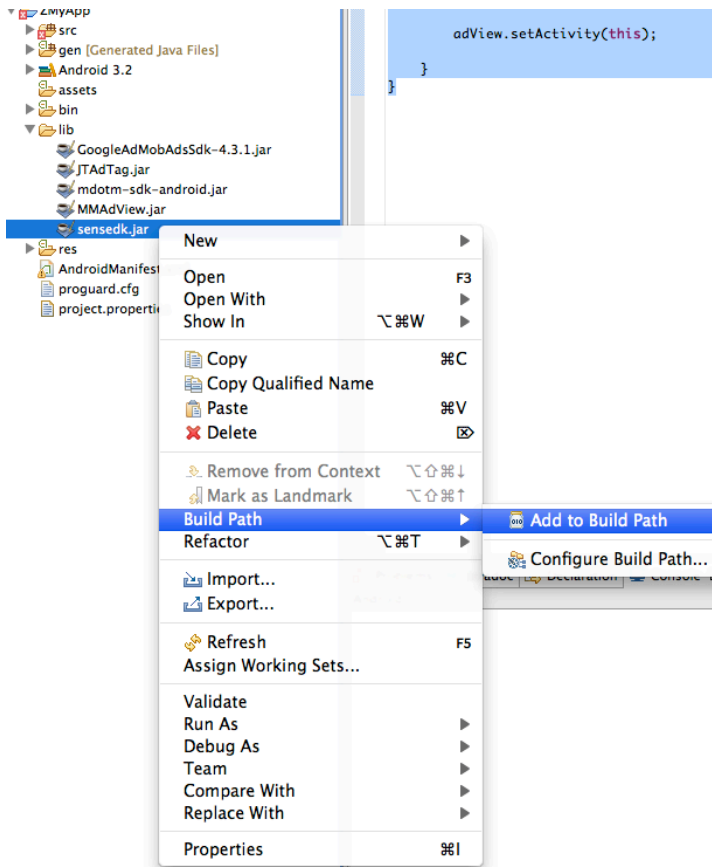
The Addience SDK currently supports four mobile ad networks as well as Sense Networks' proprietary ads. This documentation will guide you through the installation step by step, ad-network by ad-network. You can select all or a subset of the supported ad networks:

- AdMob
- Millenial Media
- JumpTap
- MdotM

1. Adding the Addience SDK to your app

1.1 Add the jar file

Add the Addience SDK JAR file (*sensedk.jar*) to your build path in your Android project. One way to do this is to copy the jar file to some location within your project and then do a right-mouse click -> Build Path -> Add to Build Path:



1.2 Edit your AndroidManifest.xml

You need to add the following highlighted sections to your own AdroidManifest.xml file. Don't forget to replace the api key (in red) with your personal api key. If you have not received your api key yet, please email support@addience.com.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.sensenetworks"
    android:versionCode="1"
    android:versionName="1.0" >
    <application
        android:icon="@drawable/ic_launcher"
```

```

        android:label="@string/app_name" >
        <activity
            android:label="@string/app_name"
            android:name=".AActivity" >
            <intent-filter >
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <meta-data android:name="com.sensenetworks.apikey"
            android:value="INSERT-YOUR-KEY-HERE" />

    </application>

    <!-- REQUIRED PERMISSIONS: -->
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />

    <!-- OPTIONAL, BUT RECOMMENDED PERMISSIONS: -->
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

</manifest>

```

1.3 Adding Banner Ads to your Layout

You can select to put ads anywhere in your layout, even in multiple layouts if you wish. We recommend to only add one ad to your layout at a time, at a position where the ads will remain visible for an extended period of time (i.e. not in splash screens). The highlighted section in the following layout example is a suggestion on how to extend your layout with an ad-view:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <com.sensedk.AswAdLayout
        android:id="@+id/myadview"
        android:layout_height="50dip"
        android:layout_width="fill_parent"
        android:layout_gravity="center"
        [ ...
        android:background="@color/adbackground"
        ...
    ]
/>

</LinearLayout>

```

1.4 Code Changes

You have to add some code in order to make Addience show ads properly. In your Activity, after loading the layout, add the highlighted lines to your onCreate(Bundle) function:

```

import android.app.Activity;
import android.os.Bundle;
import com.sensedk.AswAdLayout;

public class MyActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        AswAdLayout adView = (AswAdLayout) findViewById(R.id.adview);
        adView.setActivity(this);
    }
}

```

NOTE: We have a small 'API' section with other function calls you can add to your code at the end of this document. Some of which will allow you to control how Addience operates within your application.

2. AdMob

Follow these instructions if you choose AdMob as one of your ad networks to deliver ads. If you choose to NOT use AdMob, you may skip this section entirely.

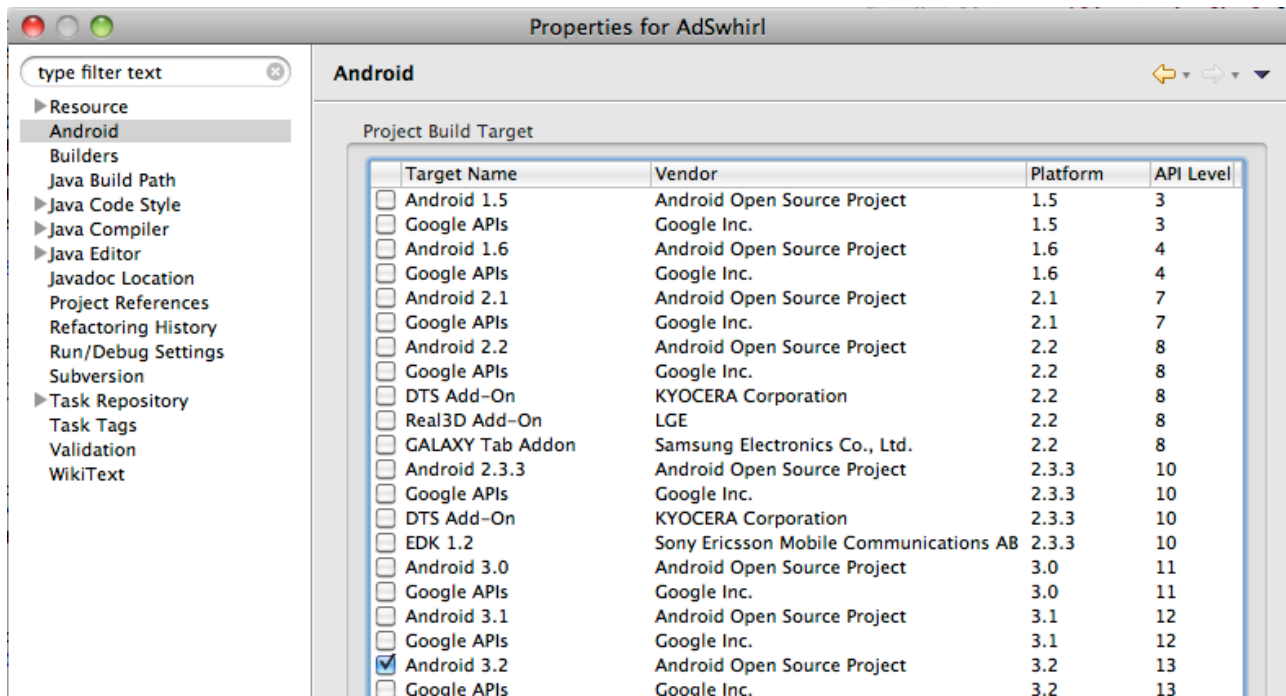
2.1 Add the jar file

Add the AdMob jar file to your Build Path as described in 1.1. You will find the jar file (GoogleAdMobAdsSdk-*.jar) in the package that shipped with the Addience SDK.

2.2 Your App and AdMob

As of version 4.3.1, AdMob is optionally using Android features from Android 3.2. This means that in order to use AdMob in your app, you have to change your Project Settings to compile with Android version 3.2 (Android-13) or higher. Unfortunately, changing your Project Settings to use Android 3.2 will prevent your app to run on devices running earlier versions of Android. To have your app support such devices, you can define a minimum version in your Android manifest file. In the following example, we'll show you how you can change your settings to use AdMob with Android 3.2 while keeping your app compatible with devices running Android 1.6 and up:

Go to Project -> Properties and select "Android" on the left. Then check Android 3.2:



Add the highlighted sections to your AndroidManifest.xml file:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.sensenetworks"
    android:versionCode="1"
    android:versionName="1.0" >

    <!-- Android Version 1.6 -->
    <uses-sdk android:minSdkVersion="4" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:label="@string/app_name"
            android:name=".MyActivity" >
            <intent-filter >
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <!-- AdMob REQUIRED Activity -->
        <activity
            android:name="com.google.ads.AdActivity"
            android:configChanges="keyboard|keyboardHidden|orientation|
                screenLayout|uiMode|screenSize|smallestScreenSize" />

    </application>
</manifest>
```

3. Millennial Media

Follow these instructions if you choose Millennial Media as one of your ad networks to deliver ads. If you choose to NOT use Millennial Media, you may skip this section entirely.

3.1 Add the jar file

Add the Millennial jar file to your Build Path as described in 1.1. You will find the jar file (MMAdView*.jar) in the package that shipped with the Addience SDK.

3.2 Your App and Millennial

Add the highlighted sections to your AndroidManifest.xml file:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.sensenetworks"
    android:versionCode="1"
    android:versionName="1.0" >

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >

        <activity android:label="@string/app_name" android:name=".MyActivity" >
            <intent-filter >
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <!-- Millennial REQUIRED Activities -->
        <activity android:name="com.millennialmedia.android.MMAdViewOverlayActivity"
            android:theme="@android:style/Theme.Translucent.NoTitleBar" />
        <activity android:name="com.millennialmedia.android.VideoPlayer"
            android:theme="@android:style/Theme.NoTitleBar.Fullscreen"
            android:configChanges="keyboard|keyboardHidden|orientation" />

    </application>

    <!-- Additional Millennial REQUIRED Permission -->
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
</manifest>
```

4. MdotM

Follow these instructions if you choose MdotM as one of your ad networks to deliver ads. If you choose to NOT use MdotM, you may skip this section entirely.

4.1 Add the jar file

Add the MdotM jar file to your Build Path as described in 1.1. You will find the jar file (mdotm-sdk-android*.jar) in the package that shipped with the Addience SDK.

4.2 Your App and MdotM

Add the highlighted sections to your AndroidManifest.xml file:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.sensenetworks"
```

```

android:versionCode="1"
android:versionName="1.0" >

<application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name" >

    <activity android:label="@string/app_name" android:name=".MyActivity" >
        <intent-filter >
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <!-- MdotM REQUIRED Activities -->
    <activity android:name="com.mdotm.android.ads.MdotmLandingPage"
        android:label="@string/app_name">
        <intent-filter>
            <category android:name="android.intent.category.BROWSABLE" />
        </intent-filter>
    </activity>

</application>
</manifest>

```

5. JumpTap

Follow these instructions if you choose JumpTap as one of your ad networks to deliver ads. If you choose to NOT use JumpTap, you may skip this section entirely.

5.1 Add the jar file

Add the JumpTap jar file to your Build Path as described in 1.1. You will find the jar file (JTAdTag*.jar) in the package that shipped with the Addience SDK.

5.2 Your App and JumpTap

Adding the jar file to your Build Path is all you have to do in order to get JumpTap ads running.

6. Proguard and Obfuscation

6. Proguard and Obfuscation

Use obfuscation with caution! By default, Proguard is integrated with the Android build system, providing code obfuscation and optimization. When you export an application (apk) from Eclipse the default settings for Proguard work well with the Addience ad mediator.

However, Addience loads classes dynamically and needs certain class names to stay the way they are. If you choose to edit your Proguard configuration to achieve additional obfuscation or optimization, you have to make sure to keep certain Addience class names untouched. You can add the following lines to your Proguard configuration to achieve exactly that:

```

-keep public class com.sensedk.** {*; }
-dontwarn com.sensedk.networks.**
-dontnote com.sensedk.networks.**

```

7. Addience SDK API

If you wish to be notified about ad network events, you can sign up your implementation of a `com.sensedk.AswAdListener` with the ad layout, using `com.sensedk.AswAdLayout.setListener(AswAdListener)`. That interface looks like the following:

```
package com.sensenetworks.sdk;
public interface AswAdListener {
    public void onAdFailedToLoad(int network);
    public void onAdReceived(int network);
    public void onAdClicked(int network);
}
```

The parameter `network` gives you information about on which of your selected ad networks the event appeared.

Other function calls are defined in this table:

in com.sensedk.AswAdLayout:	
void: setApiKey (String apikey)	<p>Set your api key. You do not have to set the api key manually if you did add your api key to the AndroidManifest as described in the documentation:</p> <pre><application ... > ... <meta-data android:name="com.sensenetworks.apikey" android:value="<your_api_key>" /></pre> <p></application></p> <p>Parameters: String:apikey Your addience api key</p>
void: nextAdNetwork ()	<p>Triggers the Mediator to skip to the next ad network for ad displays.</p>
void: setListener (AswAdListener listener)	<p>Subscribe a listener to receive notifications from this layout.</p> <p>Parameters: AswAdListener:listener An AswAdListener implementation</p>

<code>String: getNetworkName(int network)</code>	<p>Resolves the given network identifier to a network name.</p> <p>Parameters: int:network a network identifier (as handed to the AswAdListener call-back functions)</p> <p>Returns: The network name</p> <p>Throws: UnknownNetworkAdapterException</p>
<code>int: getNumberOfAdNetworks()</code>	<p>Returns the number of ad networks available for mediation.</p>
<code>void: optOut()</code>	<p>After calling this function, this device's user will no longer be receiving recommended ads for this application; we will stop to store any user-related data in our systems. Parts of user driven ad recommendations are based on collected, irrecognizable app-usage data. As part of a user-friendly policy, and to protect privacy, you (the developer) can offer this opt-out feature to your users.</p>
<code>void: optIn()</code>	<p>After calling this function, this device's user will be receiving recommended ads for this application. Parts of user driven ad recommendations are based on collected, anonymized app-usage data. As part of a user-friendly policy, and to protect privacy, you (the developer) can offer this opt-in feature to your users, in case they chose to opt-out earlier. This function is the opposite of optOut().</p>
<code>void: deleteUserData()</code>	<p>After calling this function, this device's user's data will be deleted from the ad-recommendation server systems remotely. Parts of user driven ad recommendations are based on collected, anonymized app-usage data. As part of a user-friendly policy, and to protect privacy, you (the developer) can offer this feature to your users.</p>
in com.sensedk.AswAdListener:	
<code>void: onAdNetworkSwitched(int network)</code>	<p>Invoked when the Mediator selects a network to serve ads.</p> <p>Parameters: network The id of the network that is now in use</p>
<code>void: onNoAdNetworksToSwitchTo()</code>	<p>Invoked when the Mediator cannot make use of any ad network at this point. This can have several reasons, for example a missing Internet connection, all ad networks refused to deliver ads, ...</p>

<code>void: onAdFailedToLoad(int network)</code>	<p>Invoked when the selected ad-network fails to download or display an ad.</p> <p>Parameters: network The id of the network in use</p>
<code>void: onAdReceived(int network)</code>	<p>Invoked when the selected ad-network successfully fetched an ad for display. NOTE: Not all ad-networks provide feedback about this event.</p> <p>Parameters: network The id of the network in use</p>
<code>+void: onAdClicked(int)</code>	<p>Invoked when an ad is clicked by the user. Not all ad-networks provide feedback about this event.</p> <p>Parameters: network The id of the network in use</p>

8. Legal Note

This document and the Addience SDK itself are covered under the terms of our Non-Disclosure Agreement and Terms of Service Agreement. Do not re-post or distribute the SDK or this document without our explicit permission. We are interested in your feedback on this SDK, so please let us know of any questions, concerns and/or suggestions via email at support@addience.com.