Integrating Airpush SDK
in BT 1.5 and 2.0 Android Apps

By: Sandeep Kurne

Integrating Airpush SDK in BT 1.5 and 2.0 Android Apps

In this tutorial we are going to see how to implement the Airpush Notifications in BT 1.5 and 2.0 Android apps. A video tutorial is also available on the below link.

http://youtu.be/jmsosb_12r0

After registering a new app on your air push account, download the SDK zip package file which contains:

1. Airpush JAR file
2. Read me file
3. Sample project file

There is also an option to download the SDK documentation in PDF format which shows detailed steps to integrate the Android SDK in your app.

Each SDK file is unique for the app you register so, you will need to download the new SDK file and instructions files for each separate app you create. In short same jar file cannot be used in any other project.

Step 1 – Adding the JAR

Create libs folder in your project file and then copy paste the Airpush JAR file in libs folder.

- Right-click on your project from the Package Explorer tab and select "Properties"
- Select "Java Build Path" from the left panel
- Select "Libraries" tab from the main window
- Click on "Add JARs..."
- Select the JAR that's been copied to the libs folder
- Click "OK" to add the SDK to your Android project
- Select "Order and Export" tab from the main window and check the SDK.

Step 2 – Editing Your Manifest File

First you will need to note your Airpush **<Yourappid>** which was given to you when registering your Android application. It is a numeric code and can be found by locating your app in the apps dashboard. Remember the following code should only be used as reference and the metadata, app id, api key, etc. may differ for your project since each jar file and the pdf file with instructions is uniquely created for each app seperately.

Just before the closing </application> tag of your AndroidManifest.xml file, you will need to add these things from the pdf file you receive. :

1: Copy and paste the following code just before the closing </application> tag just under where the map library is used (line 74 or something):

```
        <!--Start required declaration for all ads-->

    <meta-data android:name="com.aTcGpjC.FUClyws114666.APPID" android:value="<Yourappid>"/>
    <meta-data android:name="com.aTcGpjC.FUClyws114666.APIKEY" android:value="FUClyws114666*1347939239114666953"/>

        <!--This activity is required for all ads. This activity is also used for Dialog ads. -->
    <activity  android:exported="false"  android:name  ="com.aTcGpjC.FUClyws114666.OptinActivity"  android:configChanges="orientation|keyboardHidden"
android:theme="@android:style/Theme.Translucent"/>

        <!--Start declaration for push notification ads. -->
    <service android:name="com.aTcGpjC.FUClyws114666.PushService" android:exported="false"/>
```

```
<receiver android:name="com.aTcGpjC.FUClyws114666.BootReceiver" android:exported="false">

<intent-filter>
    <action android:name="android.intent.action.BOOT_COMPLETED"/>
    <category android:name="android.intent.catagory.HOME"/>
</intent-filter>

</receiver>

<!--End declaration for push notification ads. -->


<activity android:name="com.aTcGpjC.FUClyws114666.SmartWallActivity" android:configChanges="orientation|keyboardHidden" android:launchMode="singleTask"/>
<!--End of Airpush Manifest declaration-->
```

Remember to replace **<Yourappid>** in the android value field with your app id.

2: Add Following Permissions:

You will need to add following additional permissions in the permission field (after line 15 or something):

```
<!--Additional required permission for push notification. Apply only for push notification ad. -->

<uses-permission android:name ="android.permission.RECEIVE_BOOT_COMPLETED" />

<!-- Required permission for Icon Ad. Apply only for icon ad.-->

<uses-permission android:name="com.android.launcher.permission.INSTALL_SHORTCUT"/>

<!--Optional permission for push notification ad but highly recommended enhancing your revenue stream -->

<uses-permission android:name= "android.permission.VIBRATE"/>

<!--Optional permissions -->

<uses-permission android:name="android.permission.GET_ACCOUNTS"/>

<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
```

Step 3 – Editing your acvtivity_base.java file.

Inside onCreate() method of the activity_base.java file, add following lines :

```
Airpush airpush=new Airpush(getApplicationContext());

airpush.startPushNotification(false);
```

For banner push, we need to copy the **airpush_notify.xml** file that is included along with SDK download to project layout folder. Do not make any changes to that file. An image with **push_icon.jpg** name is required. It will be displayed with push notification on status bar. The said image goes to res--->drawable folder in case of BT 2.0 apps and BT 1.5 apps.

Note: Rename one of your app icons with **push_icon.jpg.**

Step 4 – Setting Test Mode:

While integrating Airpush SDK into your application it is recommended that you use test mode. This will help you verify that your application is working fine with integrated SDK and test ads are returned every few minutes on an emulator or real device before publishing your app to Google Play. To enable test mode, please pass **true** as parameter while calling push notification as illustrated in the code below:

```
Airpush airpush=new Airpush(getApplicationContext());

Airpush.startPushNotification(true);
```

Once you have verified SDK installation, use false to turn off the test mode.

Note: Test mode is available for Push Notification Ads only.

Important Instructions:

- If you are upgrading/updating SDK then please do not forget to clean & build the project after completing all steps successfully.
- android:textAppearance="@android:style/TextAppearance.StatusBar.EventContent" Since BT apps use Android API lower than 2.3 for development we remove the above line from **TextView** in the the layout file.