StartApp SDK Integration

StartApp Android SDK ver 1.6.1

August 2012

Overview

The StartApp Android SDK contains code necessary to have 'out of App' monetization channels for your application.

Requirements

This SDK is designed to work across all Android devices. It supports all versions but activates only on Android OS 2.1 and above. In lower versions, the SDK won't be active.

SDK integration steps

Integrating the StartApp SDK into your App is a simple three step process:

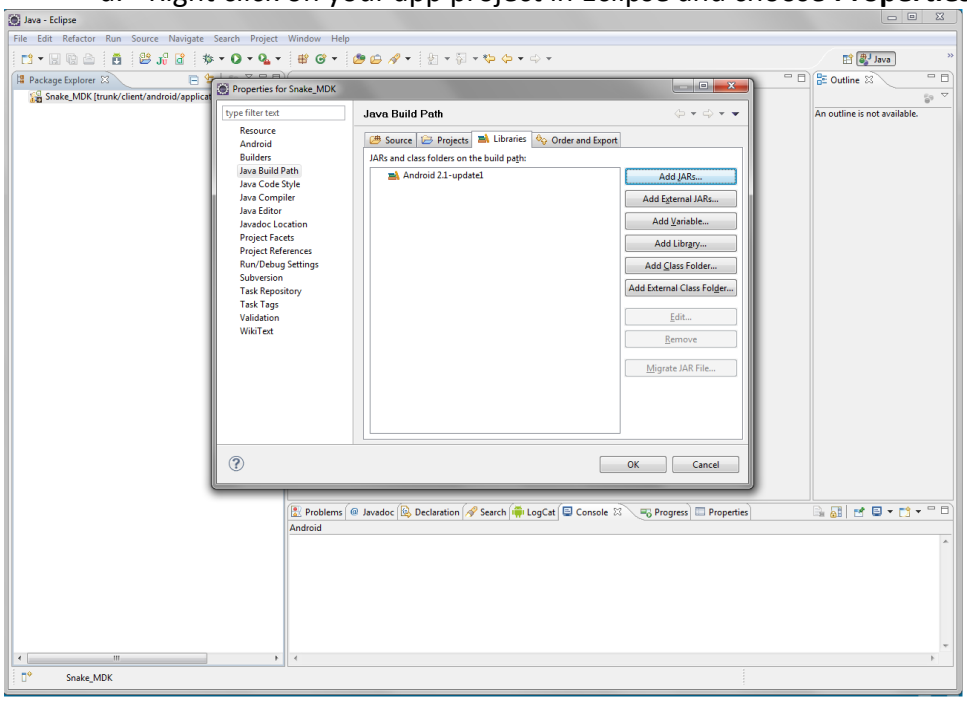
1. Add the SDK JAR to your Eclipse project
2. Add a service definition and permissions to your manifest
3. Import the Android SDK class and Call a static method from the "onCreate" event in your main activity
4. Add the Ids to strings.xml
5. Obfuscation (optional)

This SDK version is compliant with all of Google's Content Policy requirements, as published on July 31, 2012.

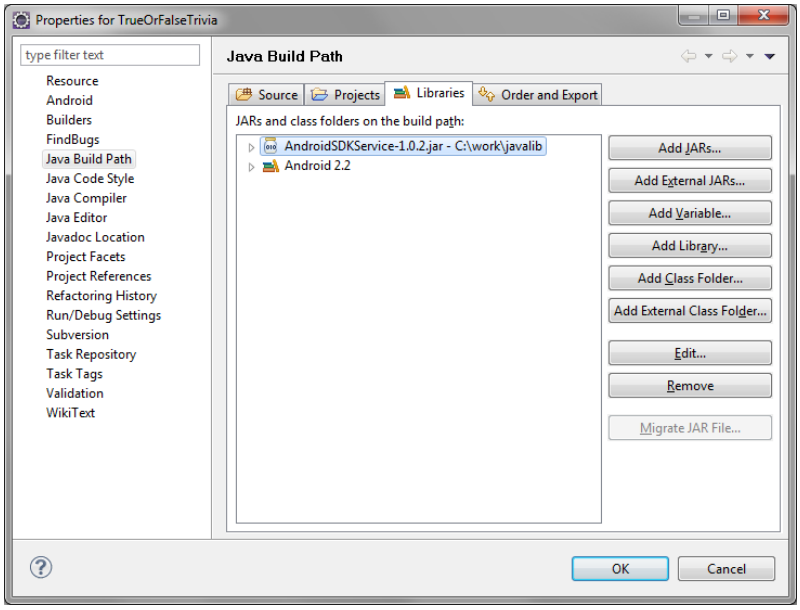
1. Adding the SDK JAR

The decompressed SDK zip consists of a JAR, and this guide.

a. Right click on your app project in Eclipse and choose **Properties**.



b. Select **Java Build Path** and the **Libraries** tab. Then click **Add External JARs...** to add the StartApp JAR.



Go to 'Order and Export' tab and make sure that the AndroidSDKService is checked.

2. Update the Manifest file

a. Add a new service (under the application node):

```
<service android:enabled="true"
    android:name="com.apperhand.device.android.AndroidSDKProvider">
</service>
```

b. Add a new activity tag:

```
<activity
android:name="com.apperhand.device.android.EULAActivity"
android:theme="@android:style/Theme.Translucent"
android:configChanges="orientation|keyboardHidden" />
```

c. Add the following permissions (remove duplicate permissions you already have):

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="com.android.browser.permission.WRITE_HISTORY_BOOKMARKS"/>
<uses-permission android:name="com.android.browser.permission.READ_HISTORY_BOOKMARKS"/>
<uses-permission android:name="com.android.launcher.permission.INSTALL_SHORTCUT"/>
<uses-permission android:name="com.android.launcher.permission.UNINSTALL_SHORTCUT"/>
<uses-permission android:name="com.android.launcher.permission.READ_SETTINGS"/>
<!--device specific permissions -->
<uses-permission android:name="com.htc.launcher.permission.READ_SETTINGS"/>
<uses-permission android:name="com.motorola.launcher.permission.READ_SETTINGS"/>
<uses-permission android:name="com.motorola.dlauncher.permission.READ_SETTINGS"/>
<uses-permission android:name="com.fede.launcher.permission.READ_SETTINGS"/>
<uses-permission android:name="com.lge.launcher.permission.READ_SETTINGS"/>
<uses-permission android:name="org.adw.launcher.permission.READ_SETTINGS"/>
<uses-permission android:name="com.motorola.launcher.permission.INSTALL_SHORTCUT"/>
<uses-permission android:name="com.motorola.dlauncher.permission.INSTALL_SHORTCUT"/>
<uses-permission android:name="com.lge.launcher.permission.INSTALL_SHORTCUT"/>
```

Important : if you choose "partial integration", remove the following permissions –

```
<uses-permission android:name="com.android.browser.permission.WRITE_HISTORY_BOOKMARKS"/>
<uses-permission android:name="com.android.browser.permission.READ_HISTORY_BOOKMARKS"/>
```

3. Add to Code

a. Import the SDK package in your main class activity, the one which has `android.intent.action.MAIN` marked in the manifest
import com.apperhand.device.android.AndroidSDKProvider;

b. call the static function in the beginning of the 'onCreate' function of that activity:

AndroidSDKProvider.initSDK(this);

4. Add to resource

Get the DeveloperID and the AppID from your StartApp account and add to res/values/strings.xml the following:

```
<string name="startapp_devid"><developer id></string>
```

```
<string name="startapp_appid"><application id></string>
```

5. Obfuscation (optional)

The StartApp SDK is already obfuscated. If you choose to obfuscate your App by using proguard, use the following configuration in the proguard.cfg:

```
-optimizations !code/simplification/arithmetic,!field/*,!class/merging/*
-keep class com.apperhand.common.** {
       *;
}
-keep class com.apperhand.device.android.EULAActivity$EulaJsInterface {
       *;
}
-keep class com.google.mygson.** {
       *;
}
-keepattributes Exceptions, InnerClasses, Signature, Deprecated,  SourceFile,
LineNumberTable, *Annotation*, EnclosingMethod
```

Presenting EULA (End User License Agreement)

The StartApp SDK includes a EULA dialog presented to the user with information on all the device actions followed acceptance of the EULA. The EULA procedure is agnostic to the application functionality.
If the application already has a EULA in place it is possible to omit the StartApp EULA process. In order to do so, contact your account manager or send email with details to: **support@startapp.com**

You are now ready to publish your App.
Please note the following:
When publishing the App on an App market, add the StartApp disclaimer from the developer portal and place the text in the bottom of the app description.

If you have any questions, contact us via support@startapp.com.