Flurry SDK

Android Upgrade Instructions

SDK version 5.3.0
Updated: 2/18/2015

Welcome to Flurry!

This file contains:

1. **Introduction**

Flurry Android SDK release versions 3.3.0 and later are modularized to allow our partners to select exactly the components they wish to integrate into their app. This will allow for more targeted updates and smaller distributions for partners that only integrate a subset of available Flurry services.

The Flurry Android Analytics library is required for any integration. You may optionally add on the Flurry Ads library. More information on these libraries are contained in their respective README documents, however a brief description follows:

- **Flurry Analytics** allows you to track the usage and behavior of your application on users' phones for viewing in the Flurry Developer Portal.
- **Flurry Advertising** allows you to earn revenue by offering advertising in your app. Supported integrations include banners, interstitials, and native ads.

3. Upgrade to 5.x from 4.x

Upgrading from version 4.x to 5.x is a straightforward process that should take less than 10 minutes. If you are upgrading from version prior to 4.0 and use ads, please first follow instructions in section 3 (upgrade to 4.0.0) and then return to this section.

Flurry Analytics

Upgrade using Android Studio:

1. Remove existing Flurry library jars from libs folder.
2. Add FlurryAnalytics_5.x.jar to your project's libs folder.

Upgrade using Eclipse:

1. Remove any existing Flurry library jars from your project's lib folder. Also, remove all reference from the Build Path.
2. Add FlurryAnalytics_5.x.jar to your project's libs folder. Right click on each JAR file and select **Build Path > Add to Build Path**.

Introducing initialize function

FlurryAgent has a new initialization function, FlurryAgent.init(), that **MUST** be called as soon as possible after your application is started. *Any FlurryAgent configuration that was previously done before a session was started should be done before init instead*. It is recommended that you call this function in the onCreate() method **of your Application class**. If you do not currently extend the Application class in your app, you can do so by adding a class such as this:

```
public class MyApplication extends Application {
  @Override
  public void onCreate() {
    super.onCreate();

    // configure Flurry
    FlurryAgent.setLogEnabled(false);

    // init Flurry
    FlurryAgent.init(this, MY_FLURRY_APIKEY);
  }
}
```

and in your AndroidManifest.xml, declaring the application to point to your new application class:

```
  <application
    android:name=".MyApplication"
    …
  </application>
```

You may also call FlurryAgent.init() just before FlurryAgent.onStartSession(). It is safe to call FlurryAgent.init() more than once provided that the same API key is always used during the lifetime of the application.

onStartSession and onEndSession

Sessions are automatically tracked on Ice Cream Sandwich (API level 14) and above. If the minSdkVersion for your app is set to 14 or greater you no longer need to manually call onStartSession() and onEndSession() unless you are tracking sessions using a Context that is not an Activity (i.e. a Service). You may safely remove these methods from your code. Apps that run on Gingerbread or Honeycomb or use context that is not an activity will still need to call onStartSession() and onEndSession() as before.

Location Tracking

Location tracking is now passive and will use cached location instead of active GPS to avoid excessive battery usage. You may call FlurryAgent.setLocation() to set the location manually if your app uses the GPS.

Modified Methods

FlurryAgent.onStartSession(Context context, String apiKey) is now deprecated. *Please use FlurryAgent.onStartSession(Context context) instead.* The API key is now supplied in the FlurryAgent.init() function and is no longer required here.

Removed Methods

FlurryAgent.getUseHttps()
FlurryAgent.setUseHttps(boolean useHttps)

HTTPS is now mandatory due to Google Play! Store restrictions. These methods have been removed from the SDK and should be completely removed from your application.

Flurry Ads

Upgrade using Android Studio:

1. Make sure to first follow steps for Flurry Analytics integration above. The Ads jar is dependent on the Analytics jar
2. Add FlurryAds5.x.jar to your project's libs folder.
3. Navigate to File -> Project Structure -> Module -> Dependencies.
4. Click the '+' button in the bottom of the 'Project Structure' popup to add dependencies.
5. Select 'File dependency' and add libs/FlurryAds5.x.jar.

Upgrade using Eclipse:

1. Make sure to first follow steps for Flurry Analytics integration above. The Ads jar is dependent on the Analytics jar
2. Add FlurryAds5.x.jar to your project's libs folder.
3. Right click on the Ads JAR file and select **Build Path > Add to Build Path**.

Code Size

The size and method count of the Flurry Ads SDK has been significantly reduced in this release. This does not affect any functionality and is code optimization.

Native Ads

The Flurry Ads SDK now includes support for Native Ads, which can be fully customized to the look and feel of your application. Please refer to the Ads README if you wish to include Native Ads in your app.

New Object API

We have introduced a new object-based API for Ads to make integration even easier than before !
It is recommended that developers switch to the new ads API as soon as possible. The FlurryAds API will continue to work as before, but has been deprecated. Also, the new API is required for Native Ads. Please see the Ads README for more details.

Please note: if you are using using Flurry to mediate a third-party SDK with ICustomAdNetworkHandler, you will need to continue using the old API to serve ads for this network.

---

3. Upgrade to 4.0.0 from 3.x

Upgrading from version 3.3.x to 4.0.0 is a straightforward process that should take less than 10 minutes. If you are upgrading from a version prior to 3.3 and use ads, please first follow instructions in section 4 (upgrade to 3.3.0) and then return to this section.

Flurry Advertising SDK now uses the Android Advertising ID provided by Google Play Services and will check for and respect the user's ad tracking preference. For more information, please visit https://developer.android.com/google/play-services/id.html

Flurry Analytics

Upgrade using Android Studio:

3. Remove existing Flurry library jars from libs folder.
4. Add FlurryAnalytics4.0.0.jar to your project's libs folder.
5. Navigate to File -> Project Structure -> Module -> Dependencies.
6. Click the '+' button in the bottom of the 'Project Structure' popup to add dependencies.
7. Select 'File dependency' and add libs/FlurryAnalytics4.0.0.jar.
8. Add Google Play Services library. Please follow instructions in http://developer.android.com/google/play-services/setup.html#Setup
9. Add v4 support library (or greater). Please follow instructions in https://developer.android.com/tools/support-library/setup.html#add-library

Upgrade using Eclipse:

3. Remove any existing Flurry library jars from your project's lib folder. Also, remove all reference from the Build Path.
4. Add FlurryAnalytics4.0.0.jar to your project's libs folder. Right click on each JAR file and select **Build Path > Add to Build Path**.
5. Add Google Play Service library. Please follow instructions at http://developer.android.com/google/play-services/setup.html#Setup
6. Add v4 support library (or greater). Please follow instructions at

https://developer.android.com/tools/support-library/setup.html#add-library

Flurry Ads

Upgrade using Android Studio:

6. Make sure to first follow steps for Flurry Analytics integration above. The Ads jar is dependent on the Analytics jar
7. Add FlurryAds4.0.0.jar to your project's libs folder.
8. Navigate to File -> Project Structure -> Module -> Dependencies.
9. Click the '+' button in the bottom of the 'Project Structure' popup to add dependencies.
10. Select 'File dependency' and add libs/FlurryAds4.0.0.jar.

Upgrade using Eclipse:

4. Make sure to first follow steps for Flurry Analytics integration above. The Ads jar is dependent on the Analytics jar
5. Add FlurryAds4.0.0.jar to your project's libs folder.
6. Right click on the Ads JAR file and select **Build Path > Add to Build Path**.

Upgrade to 4.0.0 for mediation of Google Mobile Ads (previously AdMob) Ads

1. AdMob is now Google Mobile Ads. The Google Play Services library automatically includes everything necessary for GMS to show ads. Any standalone AdMob library should be removed. Also, remove any reference to the AdMob library from the Build Path.
2. The Metadata and Activity have changed. Please remove all AdMob activities and metadata and add the following:

```
<activity
        android:name="com.google.android.gms.ads.AdActivity"
        android:configChanges="keyboard|keyboardHidden|orientation|screenLayout|uiMode|screenSize|smallestScreenSize" />
<meta-data
        android:name="com.flurry.gms.ads.MY_AD_UNIT_ID"
        android:value="@string/ffp_gms_ads_apikey" />
<meta-data
        android:name="com.flurry.gms.ads.MYTEST_AD_DEVICE_ID"
        android:value="@string/ffp_gms_ads_testdevicekey" />
<meta-data
        android:name="com.flurry.gms.ads.test"
        android:value="true" />
```

4. Upgrade to 3.3.0

Upgrading from version 3.2.x to 3.3.0 is a straightforward process that should take less than 10 minutes. If you are upgrading from a version prior to 3.2 and use ads, please first follow instructions in section 5 (upgrade to 3.2.0) and then return to this section.

Flurry Analytics

1. Remove any existing Flurry library jars from your project’s lib folder.
2. Add FlurryAnalytics3.3.x.jar to your project's libs folder.

Flurry Ads

Make sure to first follow steps for Flurry Analytics integration above. The Ads jar is dependent on the Analytics jar

Add FlurryAds3.3.x.jar to your project’s libs folder.

If you have implemented a FlurryAdListener, add the onRendered method:

```
public class MyAdListener implements FlurryAdListener {

.....

@Override
public void onRendered(String adSpaceName)
{
    // Handle rendered
}
```

---

5. Upgrade to 3.2.0

Flurry SDK releases prior to 3.2.0 included the advertising related APIs under the class FlurryAgent. Starting with version 3.2.0, advertising related methods are grouped under a new class called FlurryAds.

If you are upgrading from a previous version of the SDK, you will encounter compilation errors from javac (or Eclipse).

To fix these, at each instance of an error, simply replace references to FlurryAgent with FlurryAds.

Code example

Here’s a simple diff after upgrading to v3.2.x of the SDK from a version that pre-dates v3.2.0:

```
public class Example extends Activity implements FlurryAdListener {
      FrameLayout mBanner;
      public void onCreate(Bundle bundle) {
            super.onCreate(bundle);
            setContentView(R.layout.example);
            mBanner = (FrameLayout)findViewById(R.id.banner);
-           FlurryAgent.setAdListener(this);
+           FlurryAds.setAdListener(this);
      }
      public void onStart() {
            super.onStart();
            FlurryAgent.onStartSession(this, mApiKey);
-           FlurryAgent.fetchAd(this, mAdSpaceName, mBanner, BANNER_BOTTOM);
+           FlurryAds.fetchAd(this, mAdSpaceName, mBanner, BANNER_BOTTOM);
```

```
      }
      public void spaceDidReceiveAd(String adSpace) {
-           FlurryAgent.displayAd(this, mAdSpaceName, mBanner);
+           FlurryAds.displayAd(this, mAdSpaceName, mBanner);
      }
      public void onStop() {
            super.onStop();
            FlurryAgent.onEndSession(this);
      }
}
```

Please let us know if you have any questions. If you need any help, just email support@flurry.com!