



Analytics SDK for Android

Integration Guide – Using Android Studio

November, 2015

Subject to Change without Notice

Table of Contents

| | | |
|-------|--|---|
| 1 | General..... | 3 |
| 2 | Contents of the package | 3 |
| 3 | Integration Instructions | 3 |
| 3.1 | Adding wefiSdk Jar Files..... | 3 |
| 3.2 | Adding WEANDSF Native Libraries..... | 4 |
| 3.3 | Add key file..... | 4 |
| 3.4 | Adding Google Play Services to the Project | 5 |
| 3.5 | Adding Code to Your Application Class..... | 6 |
| 3.6 | Modifying the Android Manifest..... | 6 |
| 3.6.1 | Permissions | 6 |
| 3.6.2 | Services & receivers declaration | 7 |
| 4 | No use of undocumented features..... | 7 |
| 5 | Preserving WeFi Files | 7 |
| 6 | Open Source Libraries Used | 8 |
| 7 | 65K Methods Limitation..... | 8 |

1 General

This document accompanies the delivery of WeANDSF Developer Pack for Android Studio workspace.

2 Contents of the package

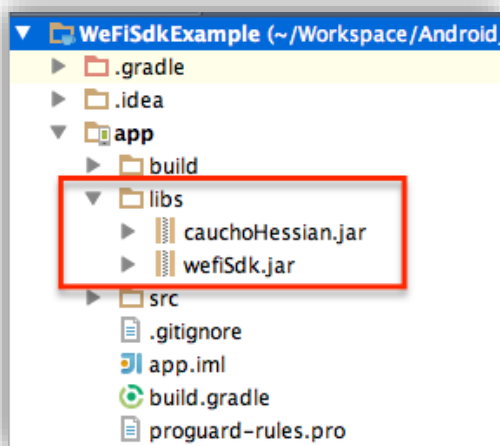
The files and folders included in the package are listed in the table below.

| File/Folder | Description | Comments |
|----------------|---|--|
| <i>libs</i> | WeFi SDK Library (Launch Candidate version) | |
| <i>jniLibs</i> | NDK library | This library is optional, Highly recommended for better performance. See details below |
| wefi.key | Unique key for your app, required for WeFi SDK initialize | |

3 Integration Instructions

3.1 Adding wefiSdk Jar Files

Add the supplied jars (cauchoHessian.jar, wefiSdk.jar) to the project libraries.

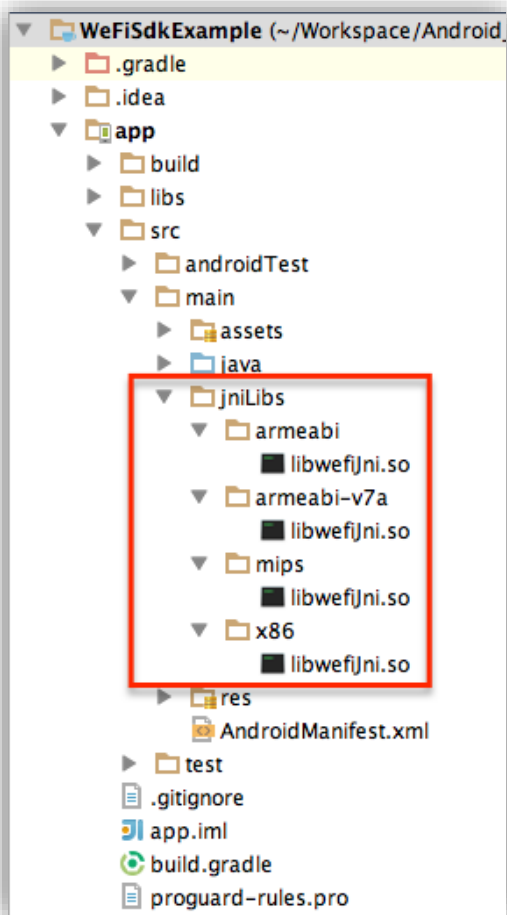


3.2 Adding JNI Libraries

Warning:

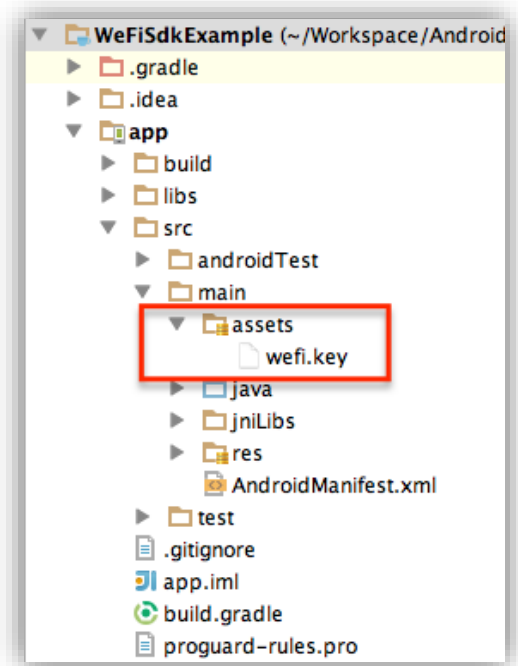
If the hosting application already uses JNI, do not add WeFi JNI library to your project to avoid run time errors for missing JNI code. Otherwise, it is highly recommended to add it for better performance of the SDK.

Copy and paste the supplied libraries: "`jniLibs\XXXX\libwefiNdk.so`" to the `app/src/main/jniLibs` folder of your main project.



3.3 Add key file

Add **wefi.key** file to assets directory under your application.

**Warning:**

If this file will not be added, WeFi SDK would not start properly and “java.io.FileNotFoundException: wefi.key” error would be written to the logs

3.4 Adding Google Play Services to the Project

Add the following line to your build.gradle file under dependencies:

```
compile 'com.google.android.gms:play-services:6.5.87'
```

Note: If the hosting application already uses google services jar with version 6.5.87 or higher, skip this step and don't add google services jar to the project.

3.5 Adding Code to Your Application Class

Inside your Application class, please add the following code **AS THE FIRST LINES** (make sure that the “super” object is the Application class itself and not a higher hierarchy class implementation of yours):

Warning:

THE FOLLOWING CODE MUST BE IN THE APPLICATION CLASS, OTHERWISE IT WILL CAUSE CRASHES DURING APPLICATION RUN TIME

```
@Override
public void onCreate() {
    super.onCreate();

    Context context = getApplicationContext();
    if (WfClient.isWefiEngineProcess(context)) {
        return;
    }
    WfClient.init(context);
    //      Add here your code for Application.onCreate if required
    //      This is done in order to prevent your code to be executed in
WfEngine   process
}
}
```

3.6 Modifying the Android Manifest

This chapter describes the changes that should be done to the Android Manifest.xml file of your main project.

3.6.1 Permissions

The followings permission should be included in the androidManifest.xml file of your MAIN project. Make sure to add them if they are not already presented.

```
<!-- Must -->
<uses-permission android:name="android.permission.INTERNET" />

<!-- Highly Recommended -->
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

<!-- Optional -->
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.GET_TASKS" />
```

3.6.2 Services & receivers declaration

The following blocks should be added to the androidManifest.xml file of your MAIN project, under the "application" section:

```
<service android:name="com.wefing.engine.WfEngineService" android:process=":wefiService" >
  <intent-filter>
    <action android:name="<App's package name>.WfEngineService" />
  </intent-filter>
</service>

<receiver android:name="com.wefing.engine.WfWatcherReceiver" android:process=":wefiService"
>
  <intent-filter>
    <action android:name="android.intent.action.BOOT_COMPLETED"/>
    <action android:name="android.intent.action.ACTION_POWER_CONNECTED" />
    <action android:name="android.intent.action.BATTERY_OKAY" />
    <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
    <action android:name="android.intent.action.USER_PRESENT" />
  </intent-filter>
</receiver>

<receiver android:name="com.wefing.engine.WfReferrerReceiver" android:exported="true"
android:process=":wefiService" >
  <intent-filter>
    <action android:name="com.android.vending.INSTALL_REFERRER"/>
  </intent-filter>
</receiver>
```

Note: Replace <App's package name> with your app's package name

4 No use of undocumented features

Only the methods and classes described in the accompanying documentation should be used. Any use of undocumented classes or methods may result in undefined and potentially damaging behavior.

5 Preserving WeFi Files

WeFi stores its configuration and other important information in the file system of the application.

The following files and folders **must not be modified or deleted** by the hosting application:

1. The folder: "/jniLibs/" and its entire contents
2. The folder: "/WeFi/" and its entire contents

The developer must make sure that these folders and files are not deleted during application upgrade process, as well as during normal operation.

Deleting any of the folders/files will result in malfunction of the API, or total stop of the SDK operation.

6 Open Source Libraries Used

The open-source libraries used by WeANDSF API are summarized in the table below:

| Library Name | Purpose | License |
|-----------------------------|--|--|
| Caucho Hessian | Implementation of the Hessian protocol | Apache Software License 1.1 http://www.apache.org/licenses/LICENSE-1.1 |
| Google Play Services | Easy access to Google services | Version 6587000 |

7 65K Methods Limitation

In you encounter the 65K Methods limitation after integrating the WeFi SDK, you can refer to the following solutions:

1. Optimize google services compilation:
Selectively compile Google Play service APIs, instead of compiling the entire package of APIs into your app

For detailed explanation, go to <https://developers.google.com/android/guides/setup#split>

Note:

WeFi SDK requires the following APIs, make sure they are included:

```
com.google.android.gms:play-services-location:7.8.0  
com.google.android.gms:play-services-ads:7.8.0
```

2. Multidex solution.

Follow instructions here: <https://developer.android.com/tools/building/multidex.html>