



AdsMOGO

Android SDK 插屏+视频

植入流程及使用方法

目 录

1. Android SDK 植入流程	1
1.1 导入广告平台的 SDK	1
1.2 添加用户权限	4
1.3 为部分广告平台声明 Activity	7
1.4 植入全屏/插屏广告	8
1.4.1 快速使用	8
1.4.2 进阶使用	10
1.4.3 视频广告	13
1.5 添加混淆配置文件（可选）	15
1.6 配置离线文件“mogoID.config”（可选）	15
2. 联系芒果	16

尊敬的开发者朋友，欢迎您使用芒果移动广告优化平台！

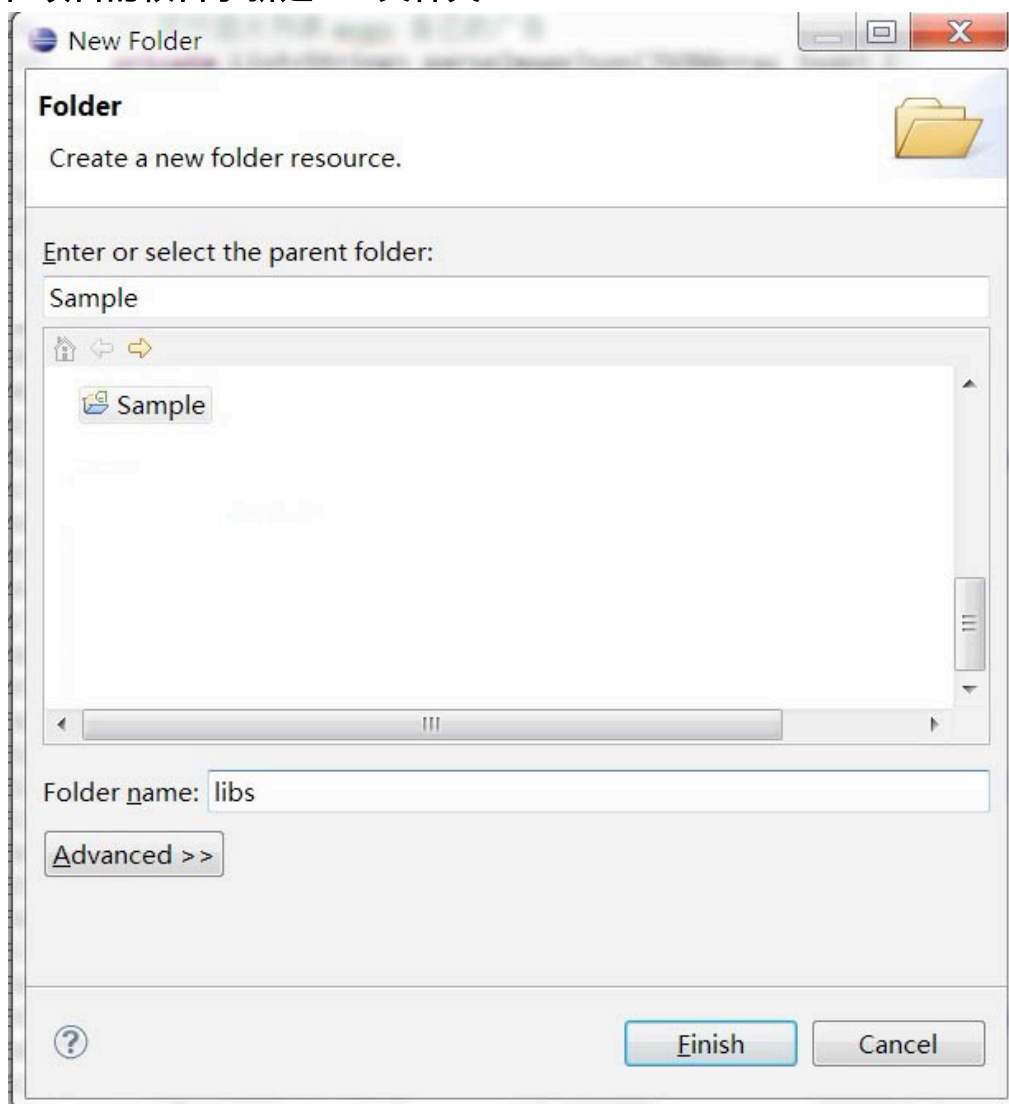
本文档为您详细介绍了AdsMOGO_Android_SDK植入流程以及使用过程中的常见问题，希望对您使用芒果有所帮助！

1. Android SDK 植入流程

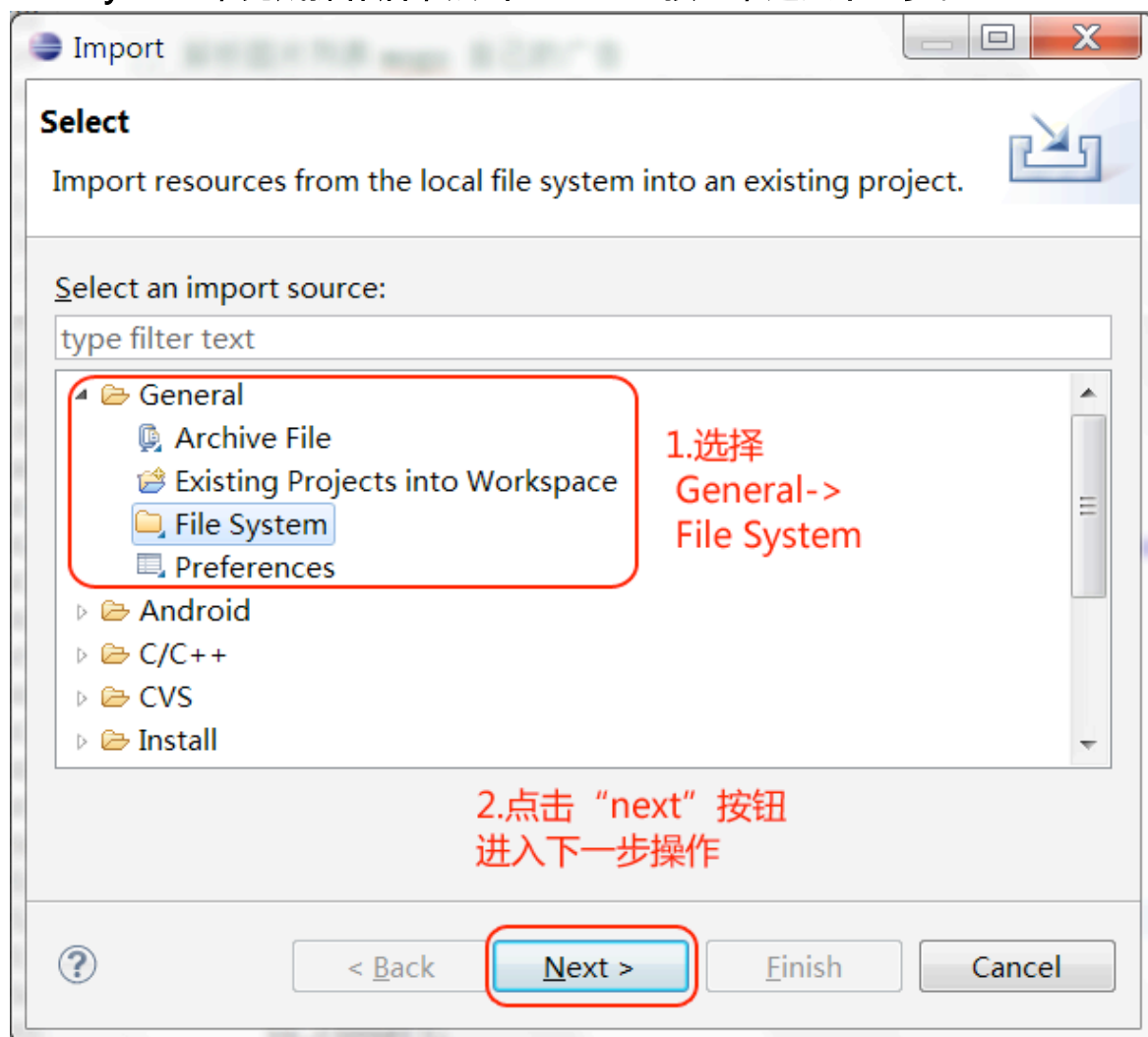
1.1 导入广告平台的 SDK

请参照以下步骤导入AdsMOGO SDK及您想要使用的各广告平台SDK。

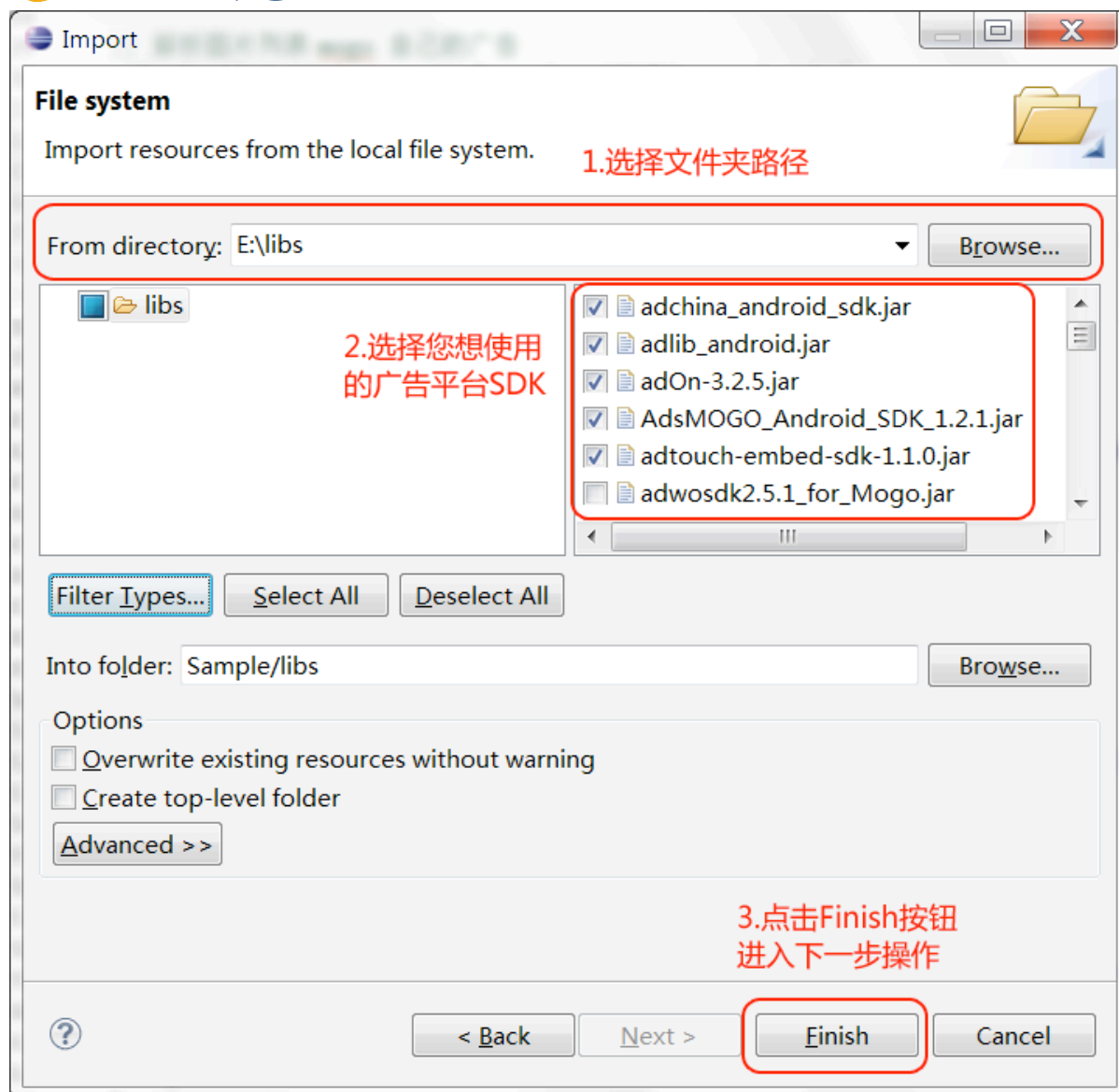
1.1.1 在项目的根目录新建libs文件夹



1.1.2 右键单击新建的libs文件夹，在弹出窗口中，选择Import -> General -> File System，完成后，点击“Next”按钮，进入下一步。



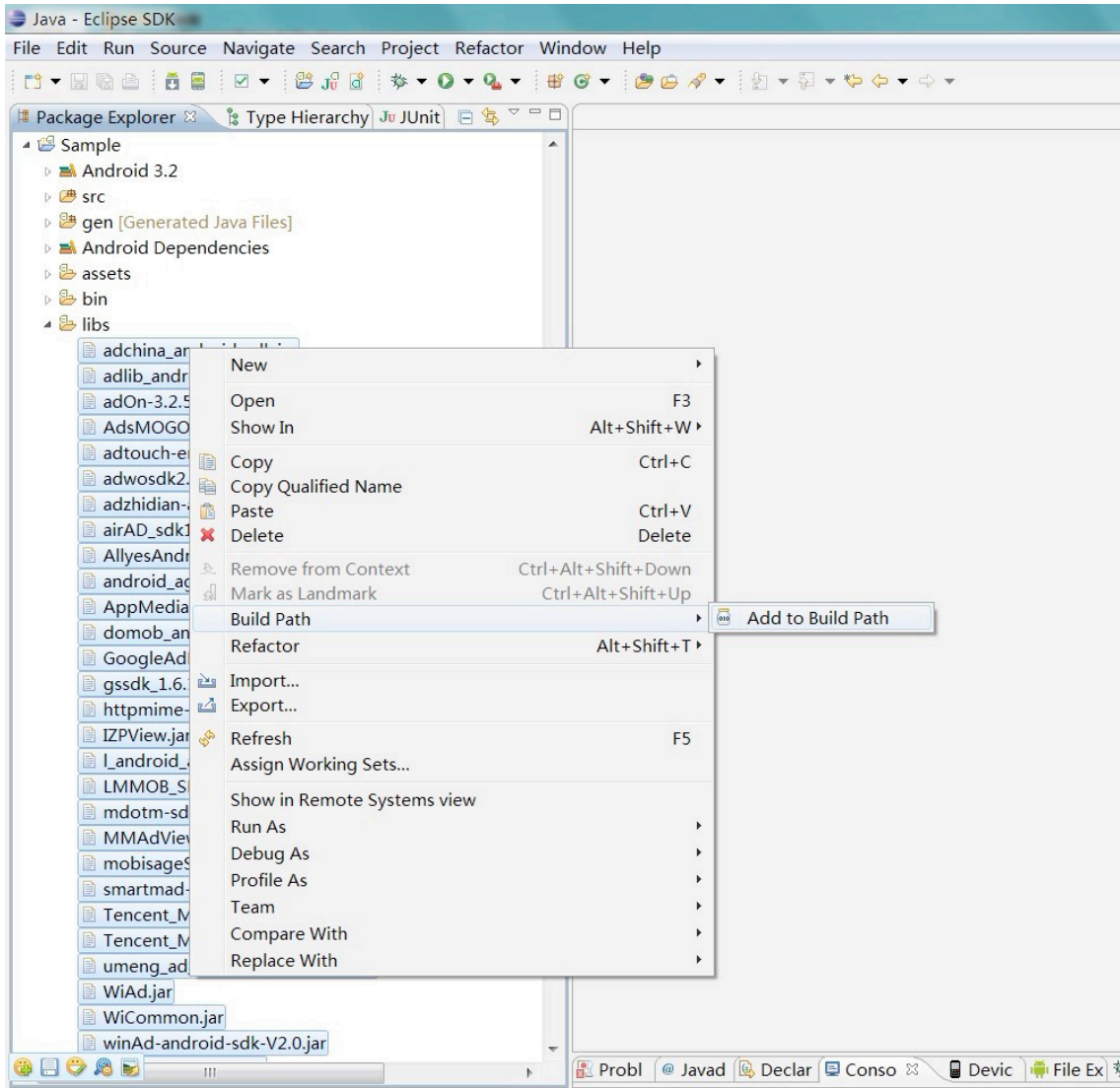
1.1.3 找到AdsMOGO_Android_SDK/libs文件夹路径，右侧选中芒果SDK (AdsMOGO_SDK_Android.jar) 和您要使用的广告平台SDK，选中Create selected folders only，点击完成按钮，如图所示：



注：关于各 jar 包对应哪个广告平台，请查看 Android SDK 下载包中“Android - jar 包对应平台列表.pdf”。

强烈建议开发者在使用芒果Android SDK v1.6.1 时，添加与该版本一同发布的单一平台SDK（Android SDK下载包中libs.zip），不要使用 1.6.0 及以前版本提供的单一平台SDK，以免为您带来不必要的麻烦。

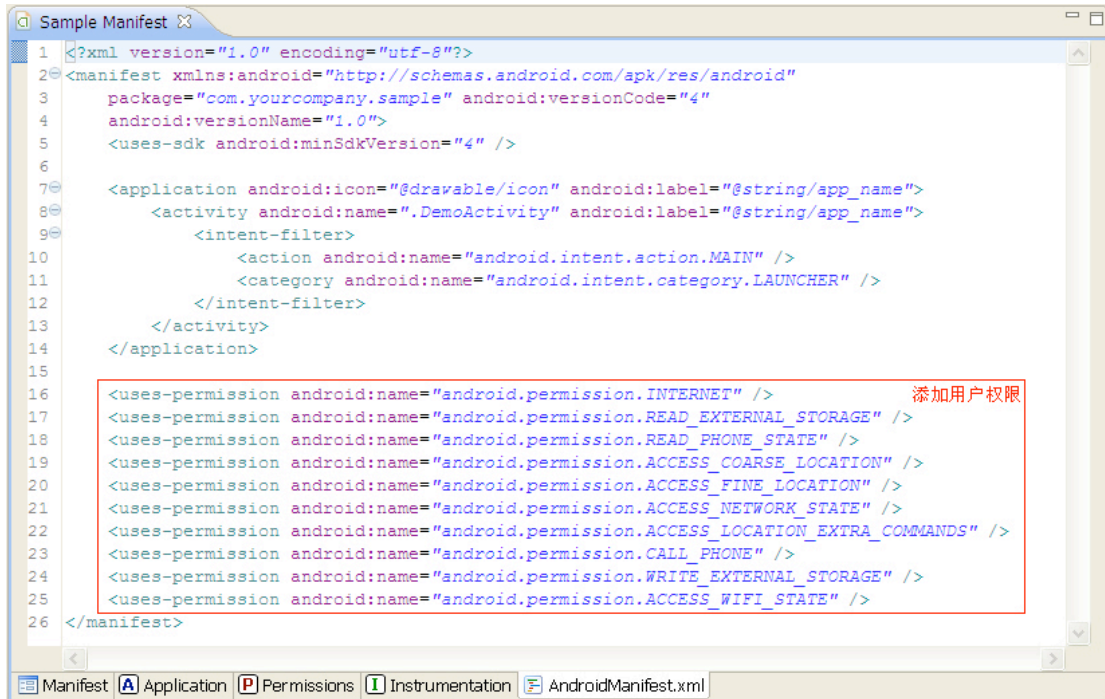
1.1.4 选中libs文件夹中所有jar包，单击右键->Build Path->Add to Build Path，如图所示：



1.2 添加用户权限

您已成功导入芒果SDK及各广告平台SDK，现在让我们开始“添加用户权限”。

如下图所示：



所需代码：（参考AdsMogoDemo/AndroidManifest.xml文件）

<!-- 添加用户权限 -->

<!-- 连接网络权限INTERNET，用于请求广告（必须），mogo会使用该权限 -->
 <uses-permission android:name="android.permission.INTERNET" />

<!-- 读取手机基本信息权限READ_PHONE_STATE，用于精确统计用户的机型等信息（必须），mogo会使用该权限 -->
 <uses-permission android:name="android.permission.READ_PHONE_STATE" />

<!-- 读取网络信息权限ACCESS_NETWORK_STATE，用于识别gprs网络的接入点的情况（必须），mogo会使用该权限 -->
 <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

<!-- 获取用户地理信息权限，用于帮助精确统计开发者的程序发布之后的用户分布情况，mogo会使用该权限-->
 <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
 <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
 <uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />

<!-- 读写sd卡的权限，用于缓存广告所用到的图片，节省流量，并可获得更好的用户体验，mogo会使用该权限-->
 <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
 <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />



```
<!--获得当前wifi的状态，微云广告平台需要此权限，mogo会使用该权限-->
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />

<!--按效果付费功能需要的权限，mogo会使用该权限 -->
<uses-permission android:name="android.permission.GET_TASKS" />

<!-- Vpon所需权限，请求访问使用照相设备-->
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera.autofocus" />

<!--MIX(智游汇) -->
<uses-permission
android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>

<!-- 宾谷 -->
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
<uses-permission android:name="android.permission.READ_LOGS"></uses-
permission>
<uses-permission android:name="android.permission.WRITE_SETTINGS"></uses-
permission>

<!-- adchina -->
<uses-permission android:name="android.permission.CHANGE_CONFIGURATION"/>

<!--拨打电话权限，请谨慎添加-- -->
<uses-permission android:name="android.permission.CALL_PHONE" />

<!-- 聚点 允许一个程序广播常用intents，请谨慎添加 -->
<uses-permission android:name="android.permission.BROADCAST_STICKY" />

<!-- 微云所需权限 此权限为开机启动，请谨慎添加-->
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />

<!-- wooboo 此权限会读取短信和联系人，请谨慎添加-->
<uses-permission android:name="android.permission.READ_SMS" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<!-- 艾德尔 可选权限 写入日程，但不可读取-->
<uses-permission android:name="android.permission.WRITE_CALENDAR" />
<!--艾德尔 可选权限 允许程序读取用户的日程信息-->
<uses-permission android:name="android.permission.READ_CALENDAR" />
<!--多盟所需权限，无此权限多盟将无法展示广告-->
<uses-permission android:name="android.permission.VIBRATE" />
```




注：您必须添加以下两个权限，否则无法获取广告。

`android.permission.INTERNET`

`android.permission.ACCESS_NETWORK_STATE`

注：当您用到下载应用功能时，请添加以下两个权限。

`android.permission.WRITE_EXTERNAL_STORAGE`

`android.permission.READ_EXTERNAL_STORAGE`

1.3 为部分广告平台声明 Activity

使用AdsMOGO SDK必须要声明MOGO的Activity。如果您使用帷千、A立方、有米、多盟、Vpon、AdMob、AdChina、AppMedia、赢告、Greystripe、MdotM、Millennial Media、AirAD、LSense、MobWIN等广告平台，您必须为它们声明Activity，才可以正常使用。

请选择添加以下代码到AndroidManifest.xml文件中的application标签中：

（参考AdsMogoDemo/AndroidManifest.xml文件）

注：建议开发者使用哪些平台，就在AndroidManifest.xml中配置哪些平台的activity、service等，以免被市场标注携带恶意广告。

```
<!-- 声明Mogo的Activity -->
<activity
    android:name="com.adsmogo.adview.AdsMogoWebView"
    android:configChanges="orientation|keyboardHidden|screenSize"
    android:label="@string/app_name" >

<!-- 声明芒果视频广告activity -->
<activity
    android:name="com.adsmogo.interstitial.video.AdsMogoVideoActivity"
    android:screenOrientation="landscape"
    android:configChanges="keyboard|keyboardHidden|orientation|screenSize" />

<!-- 注册Service，自主广告与交换广告的应用下载功能所需 -->
<service android:name="com.adsmogo.controller.service.UpdateService"

    android:exported="true"


    android:process=":remote" />

<!-- 注册Service，安装应用所用 -->
<service android:name="com.adsmogo.controller.service.CountService"

    android:exported="true"

    android:process=":remote" />
```

注：由于声明Activity的代码过多，这里只将特殊的广告平台列举出来，其它的平台可参考芒果提供的AdsMogoDemo。

* 如果您用到芒果自有广告（自主，交换，自售广告）及原生整合的广告平台（后台App配置平台中标有图标）的下载应用功能，请注册以下Service。

com.adsmogo.controller.service.UpdateService

com.adsmogo.controller.service.CountService

* 如果您使用到了Vpon广告平台，操作步骤请参考芒果提供的《Vpon帮助文档.txt》，也可参照Vpon的在线帮助文档。

* 使用Google AdMob 广告 SDK（Android 版）要求采用 Android 1.5 或更高版本。请确保您使用的是最新版本的 Android SDK，且编译所针对的至少是 Android v3.2（将 default.properties 中的 target 设置为 android-13）。

*使用adwo广告平台 3.1 版本需要在AndroidManifest.xml中的application标签中添加android:hardwareAccelerated="true"

1.4 植入全屏/插屏广告

AdsMOGO SDK1.4.7 之后，全插屏功能进入 2.0 时代，使用方式完全改变，如果您是老用户，只需要您花几分钟的时间，就可以升级到新版SDK！

1.4.1 快速使用

快速使用是指通过最少的代码即可正常请求展示全插屏广告。

(1). 全插屏自动轮换

使用方式：

```
/**
 * 初始化全插屏对象
 * 初始化之前必须设置默认的AppKey和Activity
 */
public void initInterstitial(){

    //设置AppKey
    AdsMogoInterstitialManager.setDefaultInitAppKey(mogoID);
    //设置当前Activity对象
    AdsMogoInterstitialManager.setInitActivity(this);
    //初始化(必须先设置默认的AppKey和Activity对象才能通过此方法初始化SDK)
    AdsMogoInterstitialManager.shareInstance().initDefaultInterstitial();
    //设置回调
    AdsMogoInterstitialManager.shareInstance().defaultInterstitial()

        .setAdsMogoInterstitialListener(adsMogoInterstitialListener);
}

/**
 * 进入展示时机
 * 当应用需要展示全屏广告调用interstitialShow(boolean isWait);
 * 通知SDK进入展示时机,SDK会竭尽全力展示出广告,当然由于网络等问题不能立即展示
 * 广告的,您可以通过参数isWait来控制授权SDK在获得广告后立即展示广告。
 */
public void showInterstitial(){

    AdsMogoInterstitialManager.shareInstance().defaultInterstitialShow(true);
}

/**
 * 退出展示时机
 * 如果您之前进入了展示时机,并且isWait参数设置为YES,那么在需要取消等待广告展示的
 * 时候调用方法interstitialCancel();来通知SDK
 */
```



```
public void cancelShow(){

    AdsMogoInterstitialManager.shareInstance().defaultInterstitialCancel();

}

/**
 * 改变当前栈顶的Activity对象
 * 如果当前栈顶Activity发生变化，如果未调用该方法改变Activity对象，
 * 有可能会广告无法展示或者崩溃
 */
public void changeCurrentActivity(){

    AdsMogoInterstitialManager.shareInstance().defaultInterstitial()
        .changeCurrentActivity(this);

}

/**
 * remove销毁全插屏对象
 */
public void removeInterstitial(){

    AdsMogoInterstitialManager.shareInstance().removeDefaultInterstitialInstance();

}
```

(2). 全插屏手动轮换

A. 手动轮换后台设置

请在App信息界面中，将全/插屏间隔时间设置为禁用，如下图所示：

文字广告样式

背景颜色* 

文字颜色* 

横幅刷新时间

全/插屏间隔时间

过场效果

效果增强

位置获取 ☒

B. 代码实现和自动轮代码基本一致，区别代码如下：

```
/**
 * 设置在初始化之前
 * 设置默认的全插屏手动轮模式
 * true: 手动轮, false: 自动轮
 */
AdsMogoInterstitialManager.setDefaultInitManualRefresh(true);

/**
 * 调用默认全插屏的刷新方法，注意需要全插屏初始化完成后才可以调用刷新方法。
 */
if
(AdsMogoInterstitialManager.shareInstance().containDefaultInterstitia())
{
    AdsMogoInterstitialManager.shareInstance()
        .defaultInterstitial().refreshAd();
}
```

注：调用refreshAd方法时，请确保广告初始化已完成，否则该方法不起作用。建议在广告初始化完成的回调onInitFinish()中调用刷新广告的方法。

1.4.2 进阶使用

进阶使用方式是可以让你获得广告当前展示的状态、使用多个AppKey等功能。

(1). 设置当前Activity对象



```
//设置初始化所使用的Activity对象
AdsMogoInterstitialManager.setInitActivity(this);

//当使用过程中Activity发生变化，需要调用该方法改变当前全插屏对象的Activity对象，如果当前使用的Activity已经Finish而没有调用该方法改变Activity对象，则会造成程序崩溃。
AdsMogoInterstitialManager.shareInstance().defaultInterstitial()
    .changeCurrentActivity(this);
```

(2). 全插屏广告监听

注：全/插屏广告监听 “AdsMogoInterstitialListener”

Void onInitFinish()：全插屏初始化完成后被调用。

void onShowInterstitialScreen(String adName)：全插屏广告显示在屏幕成功调用(adName为当前广告平台名称)

void onInterstitialClickAd(String adName)：当mogo记录点击时调用(Mogo服务根据此回调记录点击数，此点击是过滤后的点击，一条广告一次展示只能对应一次点击，adName为当前广告平台名称)

void onInterstitialRealClickAd(String adName)：当用户点击广告时调用(真实点击，Mogo不会根据此回调记录点击数，此点击是无过滤的点击，adName为当前广告平台名称)

void onInterstitialCloseAd(boolean isAutomaticClosing)：当全插屏关闭的时候被调用，isAutomaticClosing标示广告是自动关闭还是用户手动关闭（只限芒果广告和自动优化，SDK平台无关闭回调的会在广告展示成功的同时调用次回调）

boolean onInterstitialClickCloseButton()：该回调在用户点击关闭按钮时调用。需要返回boolean值：return false全插屏广告将被关闭，return true全插屏广告不被关闭（只限芒果广告和自动优化）

boolean onInterstitialStaleDated(String adName)：当前预加载的广告已经缓存过期时调用，需要返回boolean值：return false立即轮换请求广告，return true等待下一次进入展示机会轮换（adName为当前广告平台名称）

View onInterstitialGetView()：此回调需要返回当前请求广告Activity中一个任意View视图，如果返回的view视图有错误，有可能导致个别平台（例如安沃）的全屏广告无法展示。

getCustomEventPlatformAdapterClass(AdsMogoCustomEventPlatformEnum enumIndex): 为自定义平台使用，获取用户自定义Adapter Class。

void onVideoReward(String adName,double reward)：当激励性视频广告完成后被调用返回积分，adName为平台名称，reward为平台返回的积分。

(3). 在应用中使用多个AppKey

上面的使用方式都是基于仅适用一个AppKey而言，如果您有使用多个AppKey的需求，您需要继续往下看，其中包括：

设置Activity

设置监听

进入展示时机



离开展示时机

具体使用示例：

```
/**
 * 初始化全插屏对象
 * 初始化之前必须设置默认的AppKey和Activity
 */
public void initInterstitial(){

    //设置当前Activity对象
    AdsMogoInterstitialManager.setInitActivity(this);
    //初始化(必须先设置默认的Activity对象才能通过此方法初始化SDK)
    AdsMogoInterstitialManager.shareInstance().adsMogoInterstitialByAppKey(mo
goID);
    //设置回调
    AdsMogoInterstitialManager.shareInstance().adsMogoInterstitialByAppKey(mo
goID).setAdsMogoInterstitialListener(adsMogoInterstitialListener);
}

/**
 *进入展示时机
 *当应用需要展示全屏广告调用interstitialShow(boolean isWait);
 *通知SDK进入展示时机,SDK会竭尽全力展示出广告,当然由于网络等问题不能立即展示
 *广告的,您可以通过参数isWait来控制授权SDK在获得到广告后立即展示广告。
 */
public void showInterstitial(){

    AdsMogoInterstitialManager.shareInstance().adsMogoInterstitialByAppKey(mo
goID).interstitialShow(true);
}

/**
 *退出展示时机
 *如果您之前进入了展示时机,并且isWait参数设置为YES,那么在需要取消等待广告展示的
 *时候调用方法interstitialCancel();来通知SDK
 */
public void cancelShow(){

    AdsMogoInterstitialManager.shareInstance().adsMogoInterstitialByAppKey(mo
goID).interstitialCancel();
}

/**
 * 改变当前栈顶的Activity对象
 * 如果当前栈顶Activit发生变化,如果未调用该方法改变Activit对象,
 * 有可能会广告无法展示或者崩溃
 */
public void changeCurrentActivity(){

    AdsMogoInterstitialManager.shareInstance().adsMogoInterstitialByAppKey(mo
goID).changeCurrentActivity(this);
}
```



```
/**
 * remove销毁全插屏对象
 */
public void removeInterstitial(){

    AdsMogoInterstitialManager.shareInstance().removeInterstitialInstanceByAppKey(mogoID);

}
```

1.4.3 视频广告

SDK版本 1.4.7 之后视频广告聚合功能使用也发生了变化，简化使用流程，使用方式与全插屏非常相似

(1). 视频自动轮换

初始化：

```
//设置当前Activity对象
AdsMogoInterstitialManager.setInitActivity(this);
//设置芒果视频广告所需要的viewGroup（仅芒果广告使用）
AdsMogoInterstitialManager.setInitVideoViewGroup(viewGroup);
//初始化(必须先设置默认的AppKey和Activity对象才能通过此方法初始化SDK)
AdsMogoInterstitialManager.shareInstance().adsMogoVideoByAppKey(mogoID);
//设置回调
AdsMogoInterstitialManager.shareInstance().adsMogoVideoByAppKey(mogoID)
    .setAdsMogoInterstitialListener(adsMogoInterstitialListener);
```

进入展示时机：

```
//进入展示机会
AdsMogoInterstitialManager.shareInstance().adsMogoVideoByAppKey(mogoID)
    .interstitialShow(true);
```

离开展示时机：

```
//取消展示机会
AdsMogoInterstitialManager.shareInstance().adsMogoVideoByAppKey(mogoID)
    .interstitialCancel();
```

改变Activity对象和芒果视频所需要的ViewGroup对象：


```
//改变activity对象
AdsMogoInterstitialManager.shareInstance().adsMogoVideoByAppKey(mogoID)
    .changeCurrentActivity(this);
//改变芒果视频广告所需要的viewGroup（仅芒果广告使用）
AdsMogoInterstitialManager.shareInstance().adsMogoVideoByAppKey(mogoID)
    .changeCurrentVideoViewGroup(viewGroup);
```

销毁全插屏对象：

```
//移除并销毁视频对象
AdsMogoInterstitialManager.shareInstance().removeVideoInstanceByAppKey(mogoID);
```

(2). 视频手动轮换

A. 手动轮换后台设置

请在App信息界面中，将全/插屏间隔时间设置为禁用，如下图所示：



文字广告样式

背景颜色* #000000

文字颜色* #FFFFFF

横幅刷新时间 15秒

全/插屏间隔时间 禁用

过场效果 随机选择

效果增强 关闭

位置获取 ☒

B. 代码实现和自动轮换代码基本一致，区别代码如下：



```
/**
 * 设置在视频广告初始化之前
 * 设置视频广告手动轮换模式
 * true: 手动轮换, false: 自动轮换
 */
AdsMogoInterstitialManager.setDefaultInitManualRefresh(true);
/**
 * 调用视频广告的刷新方法, 注意需要视频广告初始化完成后才可以调用刷新方法。
 */
AdsMogoInterstitialManager.shareInstance()
    .adsMogoVideoByAppKey(mogoID).refreshAd();
```

注：调用refreshAd方法时，请确保广告初始化已完成，否则该方法不起作用。建议在广告初始化完成的回调onInitFinish()中调用刷新广告的方法。

1.5 添加混淆配置文件（可选）

芒果SDK支持基于Android2.3.3 自带的混淆机制，只适用于导出apk文件时，并且此文件只针对AdsMOGO_Android_SDK_1.6.0.jar和亿动智道SDK的混淆配置。

您可以在这里下载混淆配置文件，复制到您当前项目的根目录下。混淆的过程如下：

- 1、下载混淆文件，文件名为proguard.cfg；
- 2、把下载下来的文件拷贝到您工程的根目录下；
- 3、在default.properties文件中填写如下代码：
proguard.config=proguard.cfg
- 4、保存完成。

注：Android SDK 1.2.1 以后版本是全新的混淆配置文件，之前的文件与 v1.2.1 以后版本不兼容，请重新下载添加

1.6 配置离线文件 “mogoID.config”（可选）

功能简介：

在芒果网站配置好广告平台后，可以下载对应的广告配置离线文件，导入到工程中和芒果 SDK 配合使用。当新用户第一次使用您的应用，手机与芒果服务器信息交互异常，无法获取广告配置时，即可调用离线文件，以便实现正常展示广告。

使用方法：

(1). 配置完广告平台后，选择SDK下载，点击 [离线配置文件下载]，下载的文件格式为.config，如图所示：

App信息

广告平台

SDK下载 >

推广工具

统计分析

请保存您App的芒果ID，您需要将它复制到App的代码中

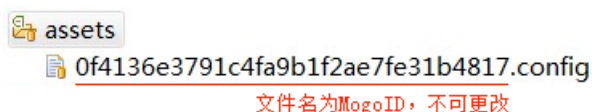
本应用的芒果ID：c255b3a879ca4b85a92592adc43786d0 [复制ID](#)

请保存您App的芒果ID，您需要将它复制到App的代码中。

为保证在断网或芒果服务器宕机等极端情况下，您的广告展示不受影响。芒果建议您在植入芒果SDK时同时植入一个“离线配置文件”。请在下方导出您的“离线配置文件”，并按照说明将其添加到App源代码工程中。

[离线配置文件下载 >>](#)

(2). 将下载的离线配置文件导入到项目**assets**文件夹中，如图所示：



恭喜您，完成了上述几个操作步骤后，您已成功植入了AdsMOGO Android SDK。现在运行您的App，就可以看到广告。接下来您可以进行测试工作，芒果完美支持真机和模拟器两种测试方式。

2. 联系芒果

如果您有任何问题或疑问，请及时联系芒果客服。我们将在第一时间做出回应。

Email：help@adsmogo.com

QQ：1346883530/1976719507

TEL：400-6088-215

芒果技术论坛经验交流版块：<http://bbs.adsmogo.com/showforum-5.aspx>