

Unity3D 集成指南

- 1. [概述](#)
- 2. [账号及相关ID准备](#)
- 3. [SDK初始化](#)
- 4. [原生广告](#)
- 5. [激励视频广告](#)
- 6. [插屏广告](#)
- 7. [Banner广告](#)

1. 概述

本文档是介绍 SDK unity3d版本的集成方法，从申请账号、appid、广告位id及集成SDK进行描述，确保开发者能够顺利集成 SDK进行变现。目前TopOn SDK的Unity 3D平台支持的广告形式如下：

广告形式	说明
Native	原生广告，无UI
Video	视频广告，带有UI
Interstitial	插屏广告，带有UI
Banner	横幅广告，带有UI

开发者可以根据应用的形态选择合适的广告形式，具体集成方法见下面具体介绍。

2. 账号及相关ID准备

你可以可通过[TopOn账号及相关ID准备](#)进行账号注册及登录的操作指引

3. SDK初始化

3.1 将TopOnSDK添加至您的项目

TopOnSDK unity 版本 你可从TopOn商务或者运营处获取，TopOnSDK包含的文件说明：（SDK是在2018.3.14f1版的基础上开发的）

文件名	描述	是否必须
anythinkunity3d.unitypackage	TopOn Unity3d 插件包,你可以导入此包到unity3d项目中,进行项目集成	是

3.1.1 Android导入说明（针对路径：/Assets/Plugins/Android 的文件说明）

1.libs插件的说明

路径	描述	是否必须
./libs/aars_anythink	TopOn的基础SDK包	是
./libs/aars_china_network	中国区聚合的第三方SDK文件夹	否
./libs/aars/aar_international_network	非中国区绝壑的第三方SDK文件夹	否
./libs/aars/aar_v4	Android Support v4+v7的SDK包（如果源项目已经有引入，则可将该文件剔除，不加入打包）	是

其中中国区的SDK文件夹和非中国区的SDK文件夹只能选择其中一个进行打包（根据开发者应用的流量来选择接入）

(1) ./libs/aars_china_network文件夹说明：

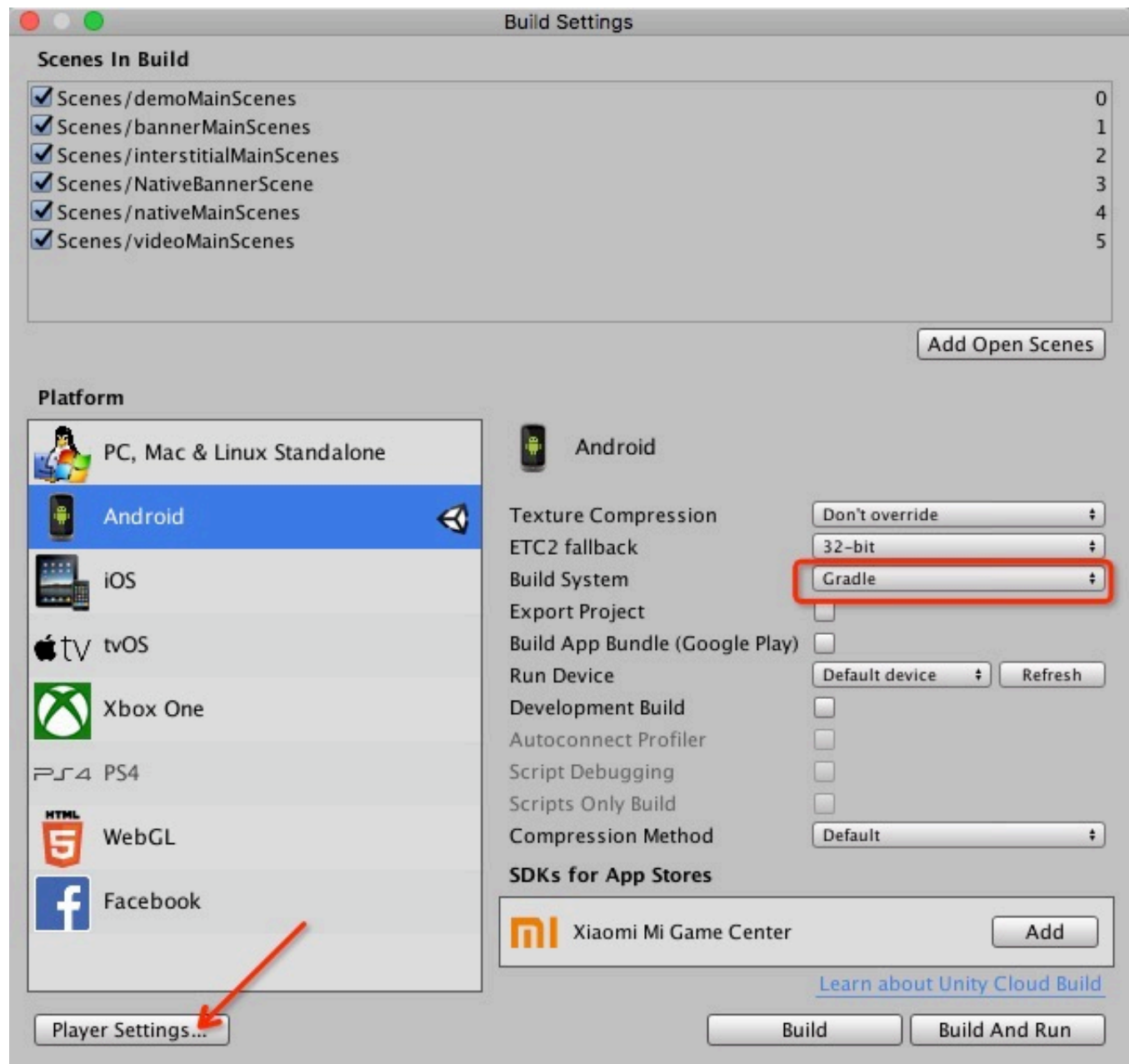
路径	描述	是否必须
./aars_china_network/aars_plugin	中国区聚合第三方SDK必须的插件包（如果源项目已经有引入，则可将该文件剔除，不加入打包）	是
./aars_china_network/aar_toutiao	穿山甲SDK文件夹	否
./aars_china_network/aar_baidu	百度SDK文件夹	否
./aars_china_network/aar_gdt	广点通SDK文件夹	否
./aars_china_network/aar_ks	快手SDK文件夹	否
./aars_china_network/aar_ksyun	金山云SDK文件夹	否
./aars_china_network/aar_luomi	洛米SDK文件夹	否
./aars_china_network/aar_mintegral_china	Mintegral（中国区） SDK文件夹	否
./aars_china_network/aar_oneway	Oneway SDK文件夹	否
./aars_china_network/aar_sigmob	Sigmob SDK文件夹	否
./aars_china_network/aar_uniplay	玩转互联SDK文件夹	否

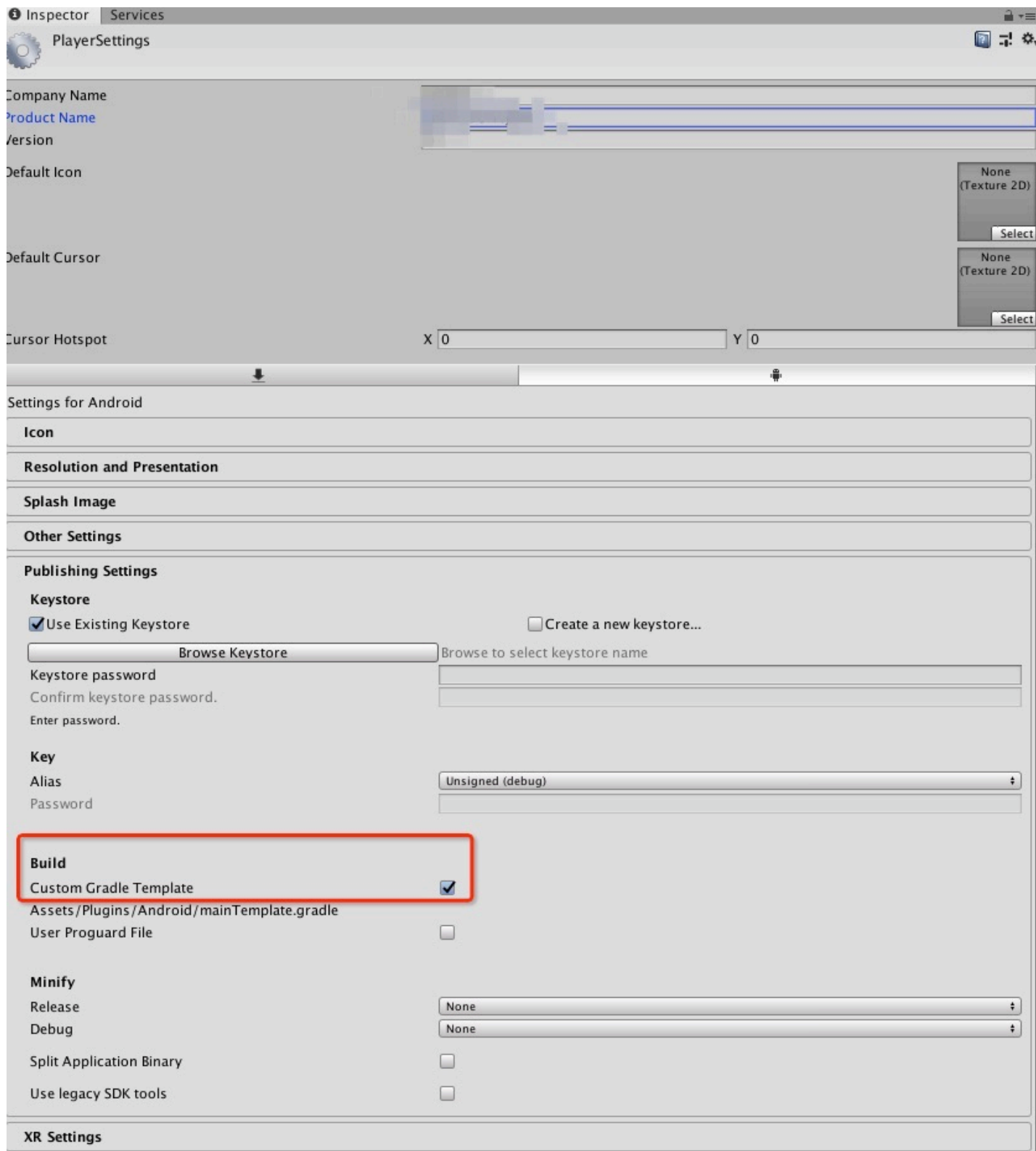
(2) ./libs/aars_international_network文件夹说明：

路径	描述	是否必须
./aars_international_network/aars_plugin	非中国区聚合第三方SDK必须的插件包（如果源项目已经有引入，则可将该文件剔除，不加入打包）	是
./aars_international_network/aars_gms_service	Google Service的SDK文件夹（如果源项目已经有引入，则可将该文件剔除，不加入打包）	是
./aars_international_network/aar_admob	Admob SDK文件夹	否
./aars_international_network/aar_facebook	Facebook SDK文件夹	否
./aars_international_network/aar_adcolony	Adcolony SDK文件夹	否
./aars_international_network/aar_applovin	Applovin SDK文件夹	否
./aars_international_network/aar_appnext	Appnext SDK文件夹	否
./aars_international_network/aar_chartboost	Chartboost SDK文件夹	否
./aars_international_network/aar_flurry	Flurry SDK文件夹	否
./aars_international_network/aar_inmobi	Inmobi SDK文件夹	否
./aars_international_network/aar_ironsource	Ironsource SDK文件夹	否
./aars_international_network/aar_maio	Maio SDK文件夹	否
./aars_international_network/aar_mintegral_international	Mintegral（非中国区） SDK文件夹	否
./aars_international_network/aar_mopub	Mopub SDK文件夹	否
./aars_international_network/aar_nend	Nend SDK文件夹	否
./aars_international_network/aar_startapp	StartApp SDK文件夹	否
./aars_international_network/aar_superawesome	SuperAwesome SDK文件夹	否
./aars_international_network/aar_tapjoy	Tapjoy SDK文件夹	否
./aars_international_network/aar_unityads	UnityAds SDK文件夹	否
./aars_international_network/aar_vungle	Vungle SDK文件夹	否

2.mainTemplate.gradle引入和说明

要先使用Unity3d来生成Android打包必须的mainTemplate.gradle文件，生成路径如下图所示：





针对生成后的mainTemplate.gradle部分说明：(SDK中有提供生成后的示例文件，由于不同版本的Unity3d工具生成的gradle文件会不一样，需要开发者删除后重新生成自身Unity3d工具下的gradle文件，SDK中的仅供参考)

```

buildscript {
    repositories {
        google()
        jcenter()
    }

    dependencies {
        //不同的Unity3d工具生成的版本号有可能会不一样
        classpath 'com.android.tools.build:gradle:3.2.0'
    }
}

...

android {
    compileSdkVersion **APIVERSION**
    buildToolsVersion '**BUILDTOOLS**'

    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }

    defaultConfig {
        minSdkVersion **MINSDKVERSION**
        targetSdkVersion **TARGETSDKVERSION**
        applicationId '**APPLICATIONID**'
        ndk {
            abiFilters **ABIFILTERS**
        }
        versionCode **VERSIONCODE**
        versionName '**VERSIONNAME**'
        multiDexEnabled true //需要额外补充设置，是为了当代码行数超过64k的时候设置的
    }
    .....
}

```

3.AndroidManifest.xml说明：

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.superapp.filemanager"
    android:versionCode="2"
    android:versionName="1.1" >

<uses-sdk
    android:minSdkVersion="16"
    android:targetSdkVersion="28"
    android:usesCleartextTraffic="true" />

<!--其中usesCleartextTraffic的配置必须设置，主要作用是让游戏里可以使用http请求（必须使用）-->
<application
    android:usesCleartextTraffic="true"
>

<activity android:name="com.unity3d.player.UnityPlayerActivity" android:label="@string/app_name">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
    <meta-data android:name="unityplayer.UnityActivity" android:value="true" />
</activity>

<!--这个设置主要是为了适配9.0以上的机器（必须使用）-->
<uses-library android:name="org.apache.http.legacy" android:required="false"/>
</application>

<!--必须要有的权限-->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

</manifest>

```

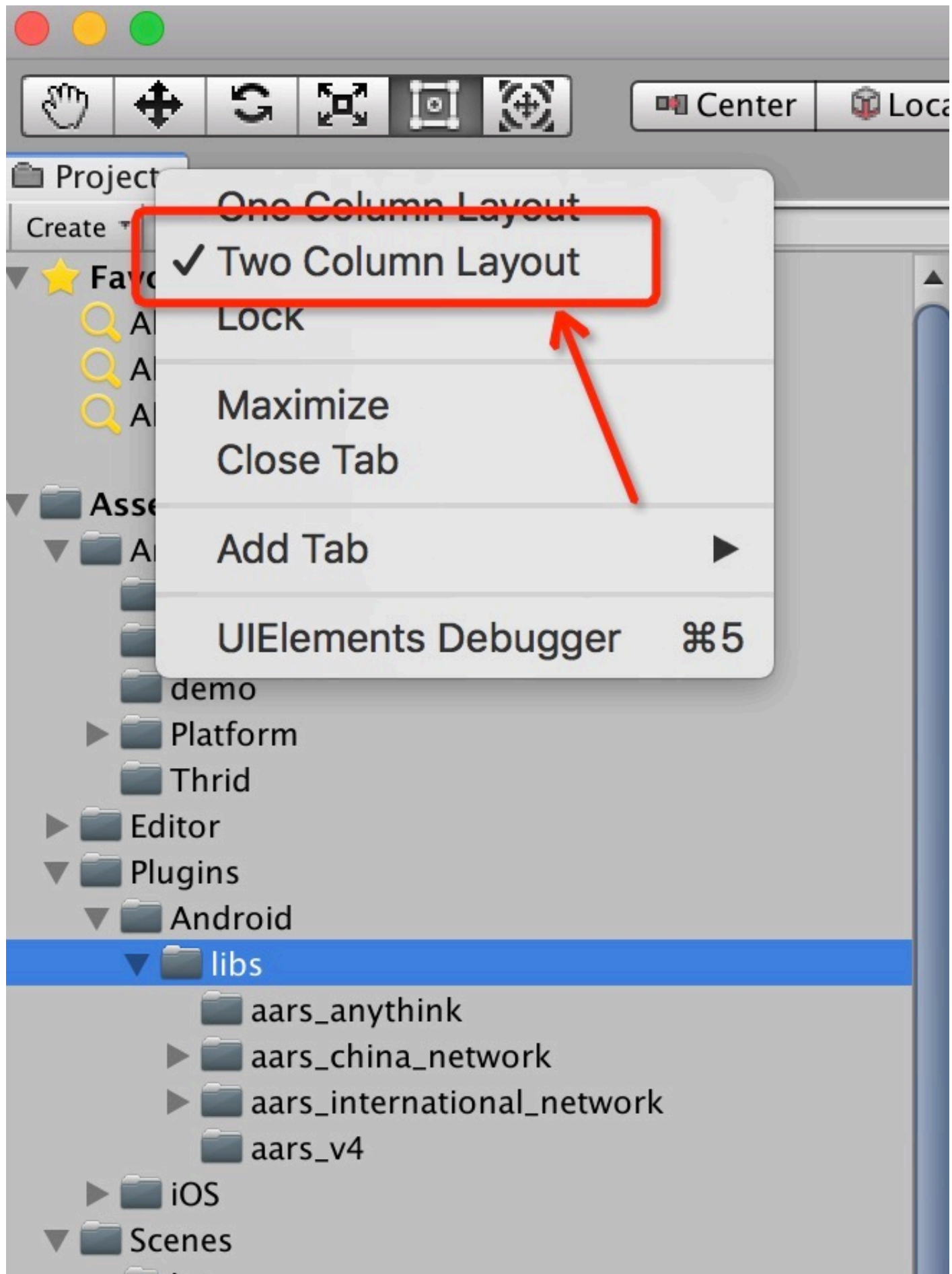
4.与其他第三方Android SDK合并的说明

- (1) 必须把第三方的jar包和aar包放在目录：/Assets/Plugins/Android/libs下
- (2) 如果第三方sdk有资源，则把资源的文件夹放在目录：/Assets/Plugins/Android/下
- (3) 如果第三方存在AndroidManifest文件，则需要和/Assets/Plugins/Android/AndroidManifest.xml文件内容合并，TopOn主要将上面说明必要的保留即可

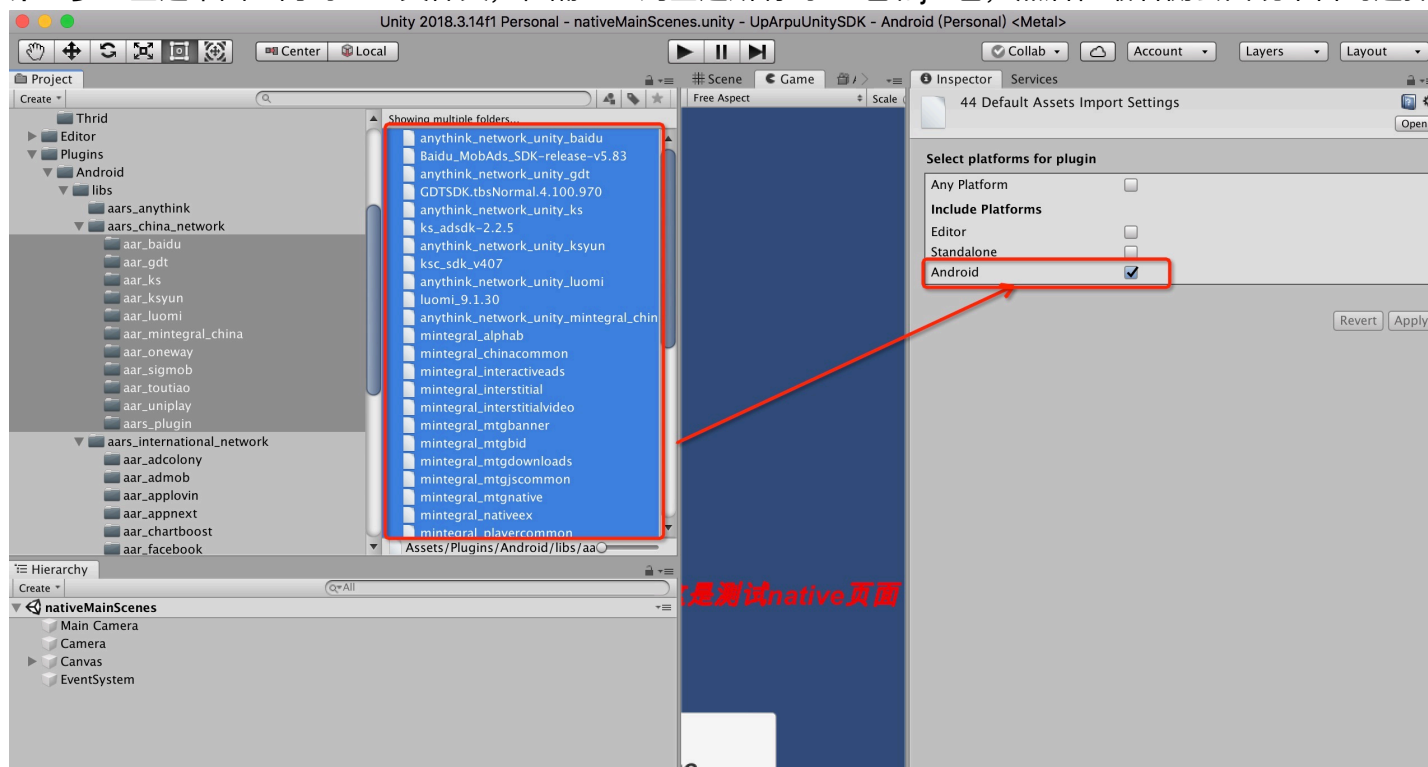
5.如何选择Android的部分AAR或者Jar不进行打包

以不引入中国区的**SDK**包为例子：

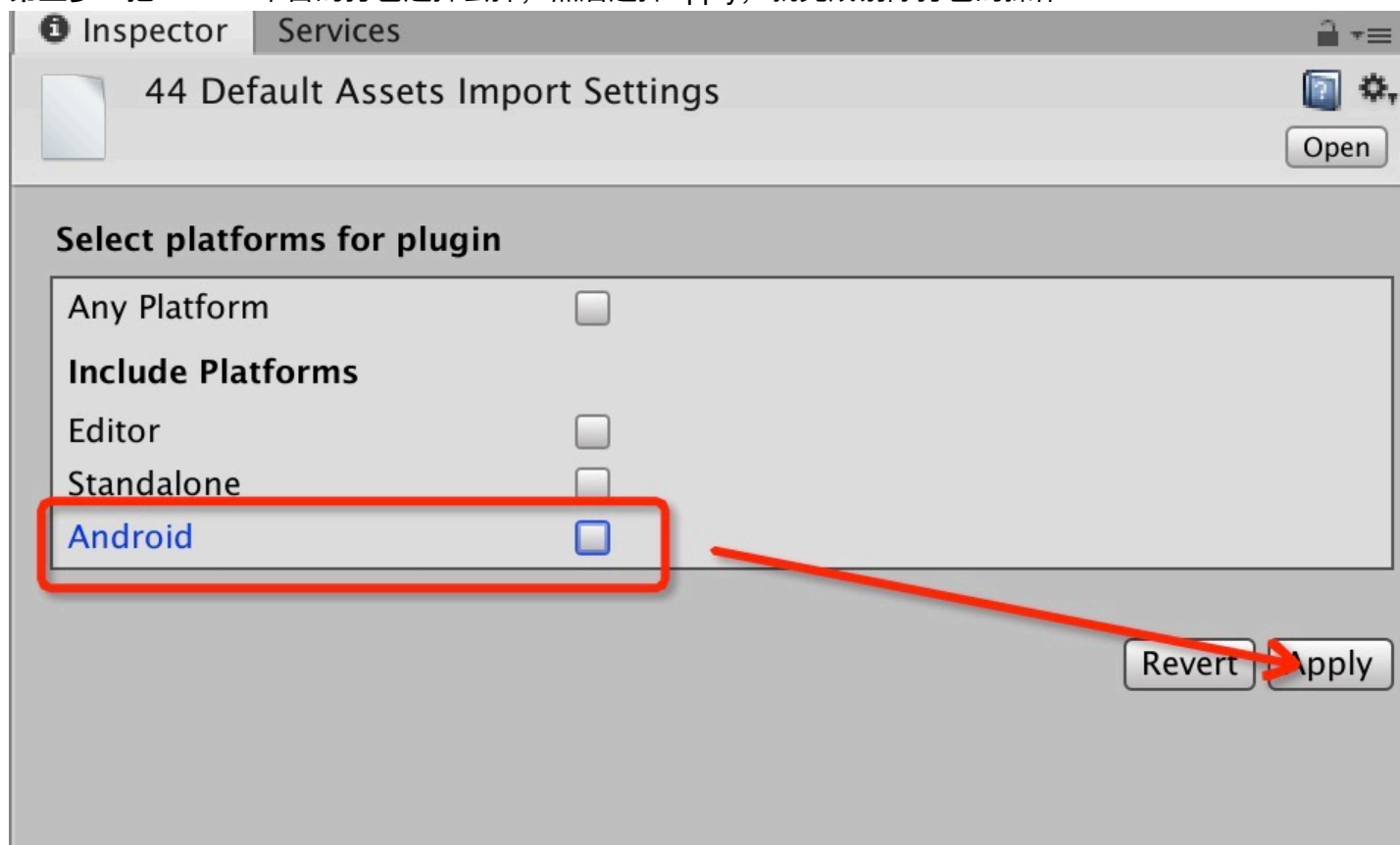
第一步： 选择可展示Unity3d中的展示文件的目录



第二步：全选中国区内的SDK文件夹，在隔壁一列全选所有的aar包和jar包，然后在最右侧会出现平台的选择



第三步：把Android平台的打包选择去掉，然后选择Apply，就完成剔除打包的操作



3.1.2 iOS平台导入说明

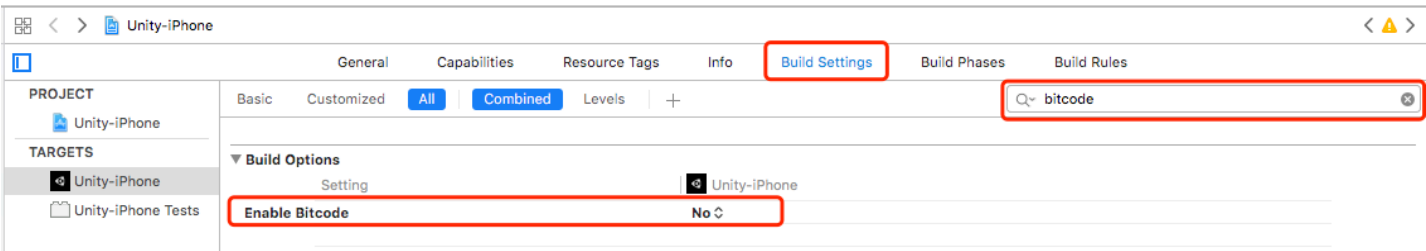
利用Unity编译出Xcode工程后，打开Xcode工程，按各第三方平台指引引入其需要的SDK并链接其依赖的系

统framework及lib等，也可以看TopOn各平台接入帮助[TopOn各平台接入帮助](#)

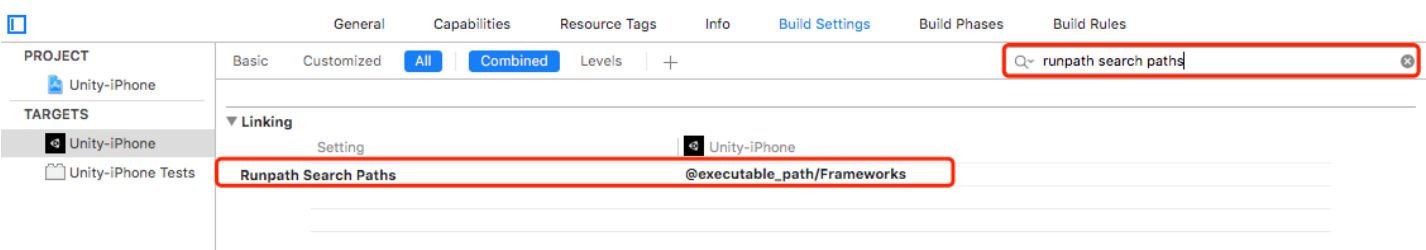
在Unity的sdk包里已经包含所有的第三方Framework包，可根据需要删除不需要的sdk包，详细哪些平台需要哪些包的引入请查看上面的帮助文档

根据以上罗列的信息引入各第三方网络所需SDK并根据各SDK要求引入系统framework和lib之后需要在Build Settings进行以下配置：

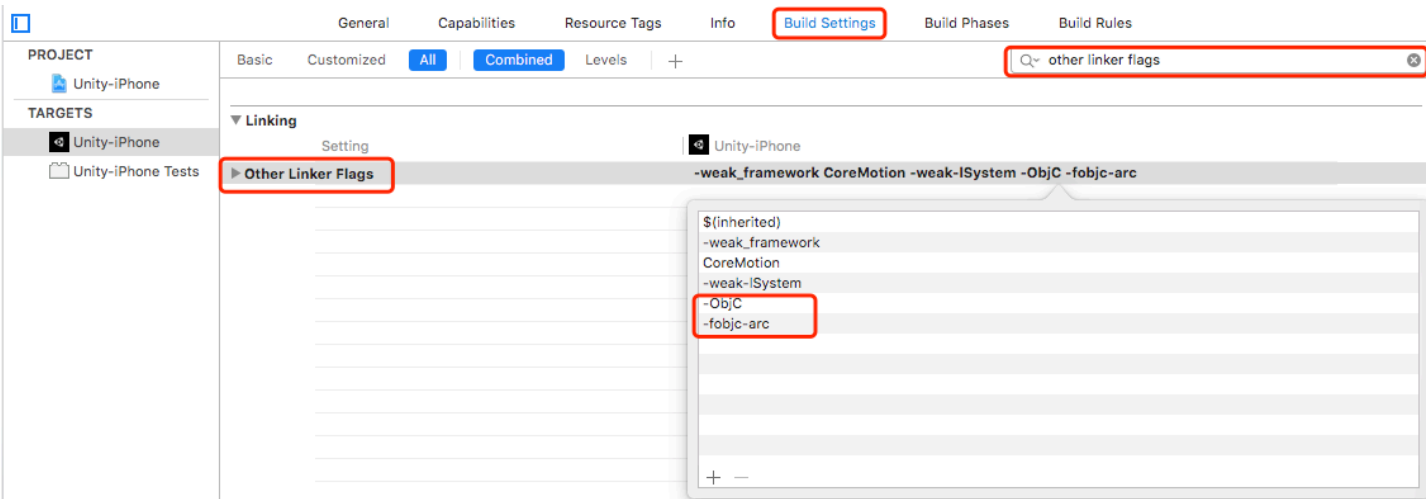
1 在Xcode 工程的Build Settings中，搜索bitcode，并将其值改为NO（当前版本Unity（2018.02）编译出来的Xcode工程中，此项设置默认为Yes），如图：



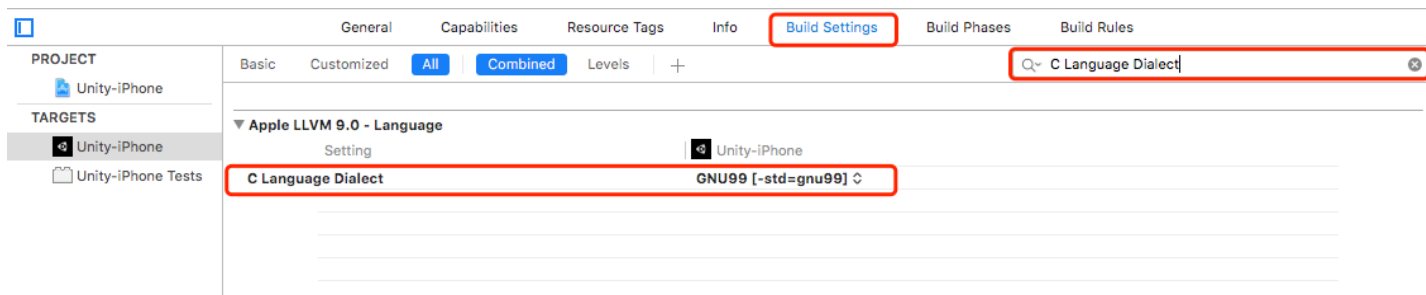
2 在Xcode 工程的Build Settings中，搜索runpath search paths，并将其值改为@executable_path/Frameworks，如图：



3 在Xcode 工程的Build Settings中，搜索other linker flags，在默认值基础上增加-ObjC, -fobjc-arc，如图：



4 在Xcode 工程的Build Settings中，搜索C Language Dialect，将其值改为GNU99[-std=gnu99]，如图：



3.2 初始化API说明

API	参数说明	功能说明
setChannel	string channel	配置SDK的渠道号信息
initCustomMap	Dictionary customMap	配置自定义参数，用于匹配后台下发广告请求列表
setLogDebug	bool isDebug	打开debug模式，用于SDK查看更多的日志
setGDPRLevel	int level	面向欧盟地区，设置GDPR隐私等级，值说明：0(完全个性化), 1(不收集设备信息,无个性化),2(禁止使用)
showGDPRAuth		展示GDPR授权页面
initSDK	string appld, string appKey	初始化SDK

3.2.1 SDK的GDPR说明

欧盟发布的《通用数据保护条例》(GDPR)将于 2018 年 5 月 25 日生效。为支持GDPR协议我们更新了 TopOn Privacy Policy，请开发者从我们官网了解 [《TopOn Privacy Policy》](#) 的相关内容。同时，为保障用户数据的隐私安全，我们在新版的TopOn SDK 2.0.0及以上中加入了数据保护功能，请开发者查阅以下文档并完成SDK接入。

1,如需要了解TopOn SDK对GDPR的详细说明，请参考[gdpr设置](#)

2,api介绍:

```

/**
 * @param level gdrp 设置隐私基本
 * 0:正常数据读取
 * 1:保持部分隐私
 * 2:完全保密,不能读取任何数据,sdk功能不能正常运行
 */
public static void setGDPRLevel(int level)

/**
 * 显示gdpr 授权页面
 */
public static void showGDPRAuth()

/**
 * 为单独平台设置gdpr设置
 * @param networkType 平台类型
 * @param dictionary 数据操作
 */
public static void addNetworkGDPRInfo(int networkType, Dictionary<string, string>
dictionary)

```

调用示例:

android:

```

//gdpr
Dictionary<string, object> dictionary ;

//admob
dictionary = new Dictionary<string, object> ();
dictionary.Add (ATConst.NEWWORK_GDPR_KEY.ADMOB_KEY_ALLOW_GDPR, "true");//是否同意gdpr
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_ADMOB,dictionary);

//inmobi
dictionary = new Dictionary<string, object> ();
dictionary.Add (ATConst.NEWWORK_GDPR_KEY.INMOBI_KEY_ALLOW_GDPR, "true");//是否同意gdpr
dictionary.Add (ATConst.NEWWORK_GDPR_KEY.INMOBI_KEY_ISGDPRSCOPE, "1");//是否gdpr地区
1:是
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_INMOBI,dictionary);

```

```
dictionary = new Dictionary<string, object> ();
//iba string
dictionary.Add (ATConst.NEWORK_GDPR_KEY.FLURRY_KEY_GDPR_IABSTR, ""); //iba串 符合iba协议的
dictionary.Add (ATConst.NEWORK_GDPR_KEY.FLURRY_KEY_ISGDPRSCOPE, "true"); //是否在gdpr地区
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_FLURRY,dictionary);

dictionary = new Dictionary<string, object> ();
dictionary.Add (ATConst.NEWORK_GDPR_KEY.APPLOVIN_KEY_ALLOW_GDPR, "true"); //是否同意gdpr
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_APPLOVIN,dictionary);

dictionary = new Dictionary<string, object> ();
dictionary.Add (ATConst.NEWORK_GDPR_KEY.MINTEGRAL_KEY_ALLOW_GDPR, "1"); //是否同意gdpr 1同意
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_MINTEGRAL,dictionary);

dictionary = new Dictionary<string, object> ();
dictionary.Add (ATConst.NEWORK_GDPR_KEY.MOPUB_KEY_ALLOW_GDPR, "true"); //是否同意gdpr
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_MOPUB,dictionary);

dictionary = new Dictionary<string, object> ();
dictionary.Add (ATConst.NEWORK_GDPR_KEY.CHARTBOOST_KEY_ALLOW_GDPR, "true"); //是否同意gdpr
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_CHARTBOOST,dictionary);

dictionary = new Dictionary<string, object> ();
dictionary.Add (ATConst.NEWORK_GDPR_KEY.TAPJOY_KEY_ALLOW_GDPR, "1"); //是否同意gdpr
dictionary.Add (ATConst.NEWORK_GDPR_KEY.TAPJOY_KEY_ISGDPRSCOPE, "true"); //是否在gdpr地区
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_TAPJOY,dictionary);

dictionary = new Dictionary<string, object> ();
dictionary.Add (ATConst.NEWORK_GDPR_KEY.IRONSOURCE_KEY_ALLOW_GDPR, "true"); //是否同意gdpr
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_IRONSOURCE,dictionary);

dictionary = new Dictionary<string, object> ();
```

```
dictionary.Add (ATConst.NEWWORK_GDPR_KEY.UNITYADS_KEY_ALLOW_GDPR, "true");//是否同意
gdpr
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_UNITYADS,dictionary);

dictionary = new Dictionary<string, object> ();
dictionary.Add (ATConst.NEWWORK_GDPR_KEY.VUNGLE_KEY_ALLOW_GDPR, "true");//是否同意gd
pr
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_VUNGLE,dictionary);

dictionary = new Dictionary<string, object> ();
dictionary.Add (ATConst.NEWWORK_GDPR_KEY.ADCOLONY_KEY_ALLOW_GDPR, "1");//是否同意gdp
r 1同意
dictionary.Add (ATConst.NEWWORK_GDPR_KEY.ADCOLONY_KEY_ISGDPRSCOPE, "true");//是否在
gdpr地区
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_ADCOLONY,dictionary);
```

以下是各平台GDPR(iOS)配置示例代码，关于各平台具体设置请查阅其官网。


```
//Admob
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_ADMOB, new Dictionary<string, object>{"consent_status":"2", "under_age":"0"});

//Inmobi
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_INMOBI, new Dictionary<string, object>{"gdpr":"0", "consent_string":"true"});

//Flurry
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_FLURRY, new Dictionary<string, object>{"scope_flag":"0", "consent_string":""});

//Applovin
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_APPLOVIN, new Dictionary<string, object>{"under_age":"0", "consent_status":"0"});

//Mintegral
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_MINTEGRAL, new Dictionary<string, object>{"0":"1"});

//Mopub
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_MOPUB, new Dictionary<string, object>{"value":"1"});

//Chartboost
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_CHARTBOOST, new Dictionary<string, object>{"value":true});

//Tapjoy
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_TAPJOY, new Dictionary<string, object>{"consent_value":"1", "gdpr_subjection":false});

//Ironsource
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_IRONSOURCE, new Dictionary<string, object>{"value":true});

//UnityAds
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_UNITYADS, new Dictionary<string, object>{"value":true});

//Vungle
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_VUNGLE, new Dictionary<string, object>{"value":1});

//AdColony
ATSDKAPI.addNetworkGDPRInfo (ATConst.NETWORK_TYPE.NETWORK_ADCOLONY, new Dictionary<string, object>{"gdpr_consideration_flag":1, "consent_string":""});
```

4. 原生广告

4.1 接入前提说明

(1)请确保已将SDK必要的包集成到工程中，并且完成AndroidManifest.xml相关权限与组件配置、混淆配置、ATSDKAPI.initSDK初始化调用。(2)需要从TopOn后台创建指定appid的native广告位ID

4.2 原生广告api说明说明

API	参数说明	功能说明
loadNativeAd	string unitId, Dictionary pairs	用于加载native广告，unitId为广告位id；pairs为空即可
setLocalExtra	Dictionary pairs	可用于设置第三方平台的本地配置
renderAdToScene	string unitId, ATNativeAdView ATNativeAdView	展示指定广告位的Native广告，ATNativeAdView为指定native广告展示的位置信息
cleanAdView	string unitId, ATNativeAdView ATNativeAdView	移除原生广告
hasAdReady	string unitId	判断指定广告位的广告是否加载完成
setListener	ATNativeAdListener listener	设置回调对象

ATNativeAdListener接口说明

API	参数说明	功能说明
onAdLoaded	string unitId	广告加载完成
onAdLoadFail	string unitId, string code, string message	广告加载失败
onAdClicked	string unitId	广告产生点击
onAdImpressed	string unitId	广告展示
onAdVideoStart	string unitId	原生视频播放开始，不同network可能支持不一样
onAdVideoEnd	string unitId	原生视频播放结束，不同network可能支持不一样
onAdVideoProgress	string unitId	原生视频播放进度，不同network可能支持不一样

调用示例:

```
//初始化请求
ATNativeAd.Instance.setListener(new ATNativeCallbackListener());

//一些平台个性化设置 如果有需求设置的话 这个例子是集成了广点通的示例
Dictionary<string,string> gdtlocal = new Dictionary<string,string>();
gdtlocal.Add ("gdtad_width","-1");
gdtlocal.Add ("gdtad_height","-1");
ATNativeAd.Instance.setLocalExtra (currunitid,gdtlocal);

//开始请求
ATNativeAd.Instance.loadNativeAd(currunitid, null);

//显示广告 — 配置
int rootbasex = 100, rootbasey = 100;
//父框架
int x = rootbasex,y = rootbasey,width = 300*3,height = 200*3,textsize = 17;
conifg.parentProperty = new ATNativeItemProperty(x,y,width,height,bgcolor,textcolor,textsize);

//adlogo
x = 0*3;y = 0*3;width = 30*3;height = 20*3;textsize = 17;
conifg.adLogoProperty = new ATNativeItemProperty(x,y,width,height,bgcolor,textcolor,textsize);

//adicon
```

```

x = 0*3;y = 50*3-50;width = 60*3;height = 50*3;textsize = 17;
config.appIconProperty = new ATNativeItemProperty(x,y,width,height,bgcolor,textcolor,textsize);

//ad cta
x = 0*3;y = 150*3;width = 300*3;height = 50*3;textsize = 17;
config.ctaButtonProperty = new ATNativeItemProperty(x,y,width,height,"#ff21bcab","#ffffff",textsize);

//ad desc
x = 60*3;y = 100*3;width = 240*3-20;height = 50*3-10;textsize = 10;
config.descProperty = new ATNativeItemProperty(x,y,width,height,bgcolor,"#777777",textsize);

//ad image
x = 60*3;y = 0*3+20;width = 240*3-20;height = 100*3-10;textsize = 17;
config.mainImageProperty = new ATNativeItemProperty(x,y,width,height,bgcolor,textcolor,textsize);

//ad title
x = 0*3;y = 100*3;width = 60*3;height = 50*3;textsize = 12;
config.titleProperty = new ATNativeItemProperty(x,y,width,height,bgcolor,textcolor,textsize);

//显示广告 二 显示
ATNativeAdView ATNativeAdView = new ATNativeAdView(config);
ATManager.ATNativeAdView = ATNativeAdView;

Debug.Log("Developer renderAdToScene--->");
ATNativeAd.Instance.renderAdToScene(currunitid, ATNativeAdView);

//清除显示区域
ATNativeAd.Instance.cleanAdView(currunitid,ATManager.ATNativeAdView);

bool isPaused;
void OnApplicationFocus(bool hasFocus)
{
    isPaused = !hasFocus;
    Debug.Log ("Developer 屏幕暂停?" + isPaused);

    ATNativeAd.Instance.onApplicationForces (currunitid,ATManager.ATNativeAdView);
}

```

```
void OnApplicationPause(bool pauseStatus)
{
    isPaused = pauseStatus;
    Debug.Log ("Developer 屏幕暂停?" + isPaused);
    ATNativeAd.Instance.onApplicationPasue (currunitid, ATManager.ATNativeAdView);
}
```

5. 激励视频广告

5.2 Video API说明

API	参数说明	功能说明
loadVideoAd	string unitId, Dictionary pairs	用于load激励视频广告，unitId为广告位id；pairs为空即可
showAd	string unitId	展示指定广告位的激励视频广告
hasAdReady	string unitId	判断指定广告位的广告是否加载完成
setListener	ATRewardedVideoListener listener	设置回调对象
setUserData	string unitId, string userId, string customData	设置奖励的用户id信息
addsetting	string unitId, Dictionary pairs	用于配置第三方平台的本地配置

接口名：ATRewardedVideoListener

API	参数说明	功能说明
onRewardedVideoAdLoaded	string unitId	广告加载完成
onRewardedVideoAdLoadFail	string unitId,string code, string message	广告加载失败
onRewardedVideoAdPlayClicked	string unitId	激励视频产生点击
onRewardedVideoAdPlayStart	string unitId	视频播放开始
onRewardedVideoAdPlayEnd	string unitId	视频播放结束
onRewardedVideoAdPlayFail	string unitId,string code, string message	视频播放失败
onRewardedVideoAdPlayClosed	string unitId, bool isReward	视频关闭, isRewarded为是否产生激励

调用示例:

```
//addsetting
//单独适配平台属性 （以下配置可不使用）
private Dictionary<string,object> addsetting(){
Dictionary<string,object> appsettinglist = new Dictionary<string,object> ();

//AdmobATRewardedVideoSetting
Dictionary<string,object> admobATRewardedVideoSetting = new Dictionary<string,object> ();
appsettinglist.Add(ATAds.Api.ATConst.NETWORK_TYPE.NETWORK_ADMOB+"", Json.Serialize(admobATRewardedVideoSetting));

//mintegralATMediationSetting
Dictionary<string,object> mintegralATMediationSetting = new Dictionary<string,object> ();
appsettinglist.Add (ATAds.Api.ATConst.NETWORK_TYPE.NETWORK_MINTEGRAL+"", Json.Serialize(mintegralATMediationSetting));

//_applovinATMediationSetting
Dictionary<string,object> _applovinATMediationSetting = new Dictionary<string,object> ();
appsettinglist.Add (ATAds.Api.ATConst.NETWORK_TYPE.NETWORK_APPLOVIN+"", Json.Serialize(_applovinATMediationSetting));
```

```

//_flurryATMediationSetting
Dictionary<string,object> flurryATMediationSetting = new Dictionary<string,object>
    ();
appsettinglist.Add (ATAds.Api.ATConst.NETWORK_TYPE.NETWORK_FLURRY+"", Json.Seriali
ze(flurryATMediationSetting));

//_inmobiATMediationSetting
Dictionary<string,object> _inmobiATMediationSetting = new Dictionary<string,object
> ();
appsettinglist.Add (ATAds.Api.ATConst.NETWORK_TYPE.NETWORK_INMOBI+"", Json.Seriali
ze(_inmobiATMediationSetting));

//_mopubATMediationSetting
Dictionary<string,object> _mopubATMediationSetting = new Dictionary<string,object>
    ();
appsettinglist.Add (ATAds.Api.ATConst.NETWORK_TYPE.NETWORK_MOPUB+"", Json.Serializ
e(_mopubATMediationSetting));

//_chartboostATMediationSetting
Dictionary<string,object> _chartboostATMediationSetting = new Dictionary<string,ob
ject> ();
appsettinglist.Add (ATAds.Api.ATConst.NETWORK_TYPE.NETWORK_CHARTBOOST+"", Json.Ser
ialize(_chartboostATMediationSetting));

//_tapjoyATMediationSetting
Dictionary<string,object> _tapjoyATMediationSetting = new Dictionary<string,object
> ();
appsettinglist.Add (ATAds.Api.ATConst.NETWORK_TYPE.NETWORK_TAPJOY+"", Json.Seriali
ze(_tapjoyATMediationSetting));

//_ironsourceATMediationSetting
Dictionary<string,object> _ironsourceATMediationSetting = new Dictionary<string,ob
ject> ();
appsettinglist.Add (ATAds.Api.ATConst.NETWORK_TYPE.NETWORK_IRONSOURCE+"", Json.Ser
ialize(_ironsourceATMediationSetting));

//_unityAdATMediationSetting
Dictionary<string,object> _unityAdATMediationSetting = new Dictionary<string,objec
t> ();
appsettinglist.Add (ATAds.Api.ATConst.NETWORK_TYPE.NETWORK_UNITYADS+"", Json.Seria
lize(_unityAdATMediationSetting));

//vungleRewardVideoSetting
Dictionary<string,object> vungleRewardVideoSetting = new Dictionary<string,object>
    ();

```

```

vungleRewardVideoSetting.Add("orientation",1);//1:2 1: 表示根据设备方向自动旋转 2:视频
广告以最佳方向播放
vungleRewardVideoSetting.Add("isSoundEnable",true);//true:false
vungleRewardVideoSetting.Add("isBackButtonImmediatelyEnable",false);//true:false 如
果为 true, 用户可以立即使用返回按钮退出广告。如果为 false, 在屏幕上的关闭按钮显示之前用户不可以使
用返回按钮退出广告
appsettinglist.Add (ATAds.Api.ATConst.NETWORK_TYPE.NETWORK_VUNGLE+", Json.Seriali
ze(vungleRewardVideoSetting));

//adColonyATRewardVideoSetting
Dictionary<string,object> adColonyATRewardVideoSetting = new Dictionary<string,obj
ect> ();

adColonyATRewardVideoSetting.Add("enableConfirmationDialog",false);//true:false
adColonyATRewardVideoSetting.Add("enableResultsDialog",false);//true:false
appsettinglist.Add (ATAds.Api.ATConst.NETWORK_TYPE.NETWORK_ADCOLONY+", Json.Seria
lize(adColonyATRewardVideoSetting));
return appsettinglist;
}

//ttATRewardedVideoSetting
Dictionary<string,object> ttATRewardedVideoSetting = new Dictionary<string,object>
();
ttATRewardedVideoSetting.Add("requirePermission",true);//是否申请权限
ttATRewardedVideoSetting.Add("orientation",1);//可选参数 设置期望视频播放的方向
ttATRewardedVideoSetting.Add("supportDeepLink",true);//可选参数 设置是否支持deeplink
ttATRewardedVideoSetting.Add("rewardName", "金币");//可选参数 励视频奖励的名称, 针对激励视
频参数
ttATRewardedVideoSetting.Add("rewardCount",1);//可选参数 激励视频奖励个数

appsettinglist.Add (ATAds.Api.ATConst.NETWORK_TYPE.NETWORK_TOUTIAO+", Json.Serial
ize(ttATRewardedVideoSetting));

//初始化
ATRewardedVideo.Instance.setListener(new ATCallbackListener());
ATRewardedVideo.Instance.addsetting (currunitid,addsetting());

//请求广告
ATRewardedVideo.Instance.loadVideoAd(currunitid,null);

```



```
//展示展示广告
ATRewardedVideo.Instance.showAd(currunitid);
```

6. 插屏广告

Interstitial初始化说明

API	参数说明	功能说明
loadInterstitialAd	string unitId, Dictionary pairs	用于load插屏广告，unitId为广告位id；pairs为空即可
showInterstitialAd	string unitId	展示指定广告位的插屏广告
hasInterstitialAdReady	string unitId	判断指定广告位的广告是否加载完成
setListener	ATInterstitialAdListener listener	设置回调对象

ATInterstitialAdListener说明

API	参数说明	功能说明
onInterstitialAdLoad	string unitId	广告加载完成
onInterstitialAdLoadFail	string unitId, string code, string message	广告加载失败
onInterstitialAdClick	string unitId	广告产生点击
onInterstitialAdShow	string unitId	广告展示
onInterstitialAdClose	string unitId	广告关闭
onInterstitialAdStartPlayingVideo	string unitId	视频播放开始，不同的network不一定存在该回调
onInterstitialAdEndPlayingVideo	string unitId	视频播放完成，不同的network不一定存在该回调
onInterstitialAdFailedToPlayVideo	string unitId, string code, string message	视频播放失败

调用示例:

```
//初始化
ATInterstitialAd.Instance.setListener(new ATInterstitialAdListener());

//请求广告
ATInterstitialAd.Instance.loadInterstitialAd(mPlacementId_interstitial_all, null);

//展示展示广告
ATInterstitialAd.Instance.showInterstitialAd(mPlacementId_interstitial_all);
```

7. Banner广告

Banner初始化说明

API	参数说明	功能说明
loadBannerAd	string unitId, Dictionary pairs	用于加载banner广告， placementId为广告位id； pairs为空即可
showBannerAd	string unitId, ATRect rect	展示指定广告位的banner广告， parameters为指定banner展示的x坐标、y坐标、w宽、h高
showBannerAd	string unitId	恢复展示banner
cleanBannerAd	string unitId	移除banner广告
hideBannerAd	string unitId	隐藏banner广告
setListener	ATBannerAdListener listener	设置回调对象

ATBannerAdListener说明

API	参数说明	功能说明
onAdLoad	string unitId	广告加载完成
onAdLoadFail	string unitId, string code, string message	广告加载失败
onAdClick	string unitId	广告产生点击
onAdImpress	string unitId	广告展示
onAdClose	string unitId	广告关闭
onAdAutoRefresh	string unitId	广告自动刷新
onAdAutoRefreshFail	string unitId, string code, string message	广告自动刷新失败

调用示例:

```
//初始化
ATBannerAd.Instance.setListener(new ATBannerAdListener());

//请求广告
ATBannerAd.Instance.loadBannerAd(mPlacementId_native_all, null);

//展示广告
ATBannerAd.Instance.showBannerAd(mPlacementId_native_all, arpuRect);
```