

Unity3D Integration

1. Introduction
2. Request account and app id
3. Import the SDK
4. Initialize the SDK
5. Integrate Native Ad
6. Integrate Rewarded Video Ad
7. Integrate Interstitial Ad
8. Integrate Banner Ad
9. On Splash Ad

1. Introduction

This document describes how to integrate the TopOn unity3d SDK from the account application to SDK integration. Following the workflow, you can integrate the TopOn SDK successfully. TopOn supports ad forms as follows in Unity 3D platform:

Ads Form	Introduction
Native	Native Ads only provides the Ads data without UI.
Video	Video Ads, with UI
Interstitial	Interstitial Ads, with UI
Banner	Banner Ads, with UI

You can choose the appropriate Ads forms of advertising according to your App's demands, the specific integration method see the following description.

2. Request account and app id

Refer to [Get Started](#) for Account Registration and Login Instructions

3. Import the SDK

3.1 SDK Package Contents

AnyThinkSDK's package contains the following contents:

File Name	Description	Necessary
anythinkunity3d.unzippackage	AnyThink Unity3d Plugin package, you can import this package into the unity3d project for integration	Yes

Note: Use Unity3D 2018 or newer for compatibility.

3.1.1 Android import introduction (Introduction for Path: /Assets/Plugins/Android)

1.libs plugin introduction

Path	Description	Necessary
./libs/aars_anythink	Core SDK of TopOn	Yes
./libs/aars_china_network	Network SDK of China	No
./libs/aars/aar_international_network	Network SDK of Non-China	No
./libs/aars/aar_v4	Android Support v4+v7 (No adding packaging if it existed in project)	Yes

The SDK folder in China and the SDK folder in non-China zone can only be selected for packaging (Select access based on the traffic of the developer application)

(1) ./libs/aars_china_network:

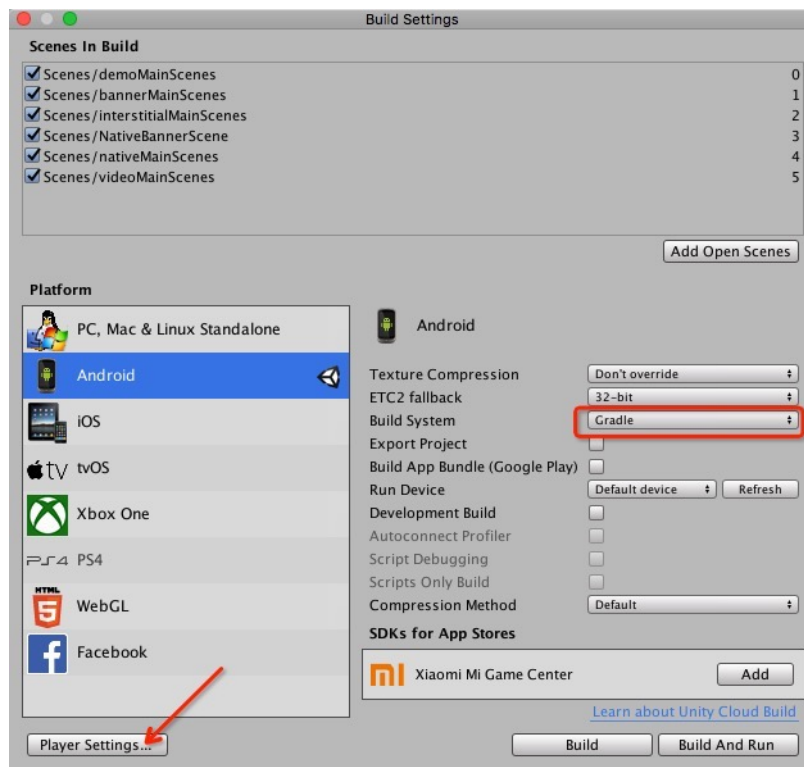
Path	Description	Necessary
./aars_china_network/aars_plugin	SDK Plugins of China (No adding packaging if it existed in project)	Yes
./aars_china_network/aar_toutiao	穿山甲 SDK	No
./aars_china_network/aar_baidu	百度 SDK	No
./aars_china_network/aar_gdt	广点通 SDK	No
./aars_china_network/aar_ks	快手 SDK	No
./aars_china_network/aar_ksyun	金山云 SDK	No
./aars_china_network/aar_luomi	洛米 SDK	No
./aars_china_network/aar_mintegral_china	Mintegral (China) SDK	否
./aars_china_network/aar_oneway	Oneway SDK	No
./aars_china_network/aar_sigmob	Sigmob SDK	No
./aars_china_network/aar_uniplay	玩转互联 SDK	No

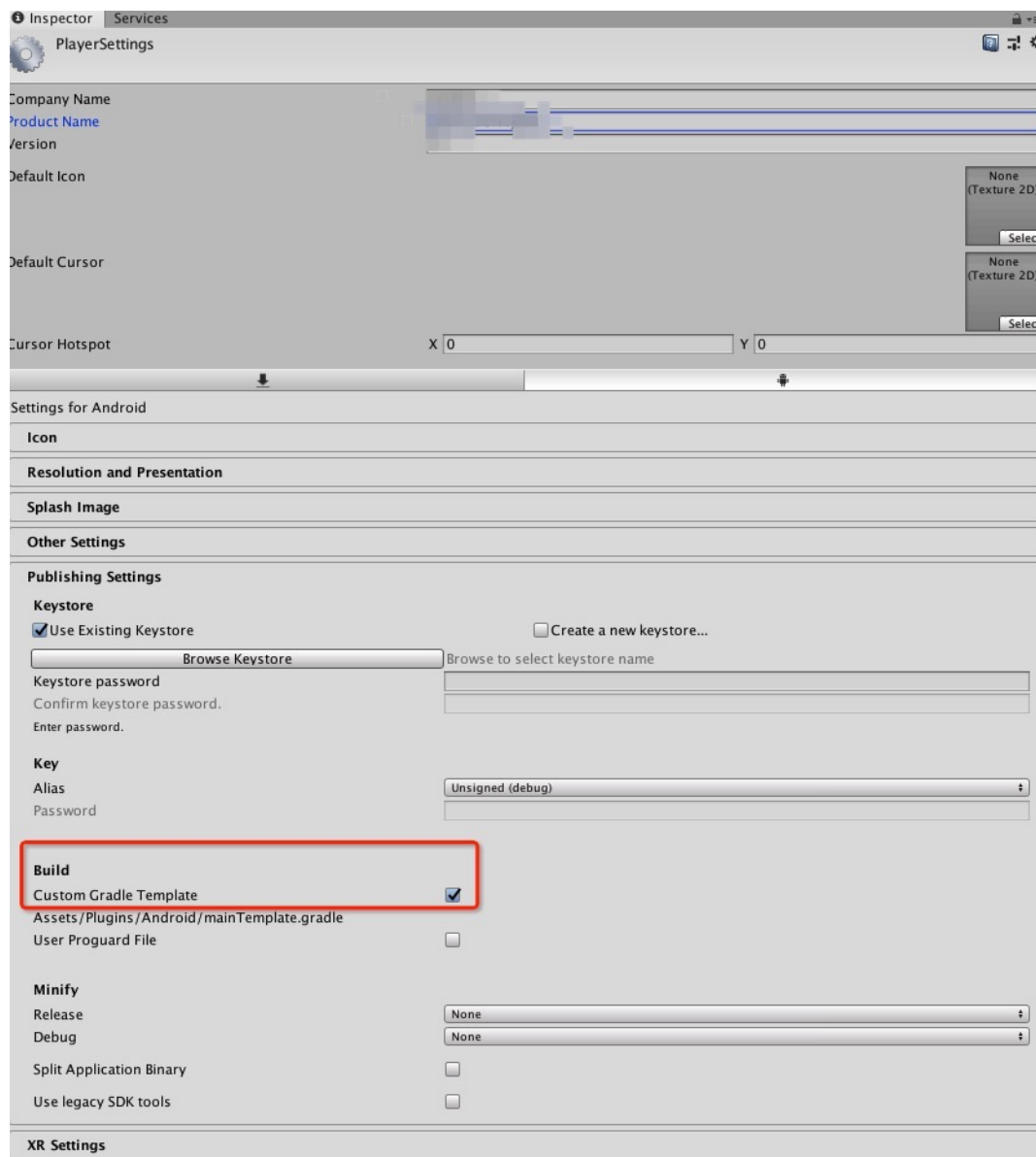
(2) ./libs/aars_international_network:

Path	Description	Necessary
./aars_international_network/aars_plugin	SDK Plugins of Non-China (No adding packaging if it existed in project)	Yes
./aars_international_network/aars_gms_service	Google Service SDK (No adding packaging if it existed in project)	Yes
./aars_international_network/aar_admob	Admob SDK	No
./aars_international_network/aar_facebook	Facebook SDK	No
./aars_international_network/aar_adcolony	Adcolony SDK	No
./aars_international_network/aar_applovin	Applovin SDK	No
./aars_international_network/aar_appnext	Appnext SDK	No
./aars_international_network/aar_chartboost	Chartboost SDK	No
./aars_international_network/aar_flurry	Flurry SDK	No
./aars_international_network/aar_inmobi	Inmobi SDK	No
./aars_international_network/aar_ironsource	Ironsource SDK	No
./aars_international_network/aar_maio	Maio SDK	No
./aars_international_network/aar_mintegral_international	Mintegral (Non-china) SDK	No
./aars_international_network/aar_mopub	Mopub SDK	No
./aars_international_network/aar_nend	Nend SDK	No
./aars_international_network/aar_startapp	StartApp SDK	No
./aars_international_network/aar_superawesome	SuperAwesome SDK	No
./aars_international_network/aar_tapjoy	Tapjoy SDK	No
./aars_international_network/aar_unityads	UnityAds SDK	No
./aars_international_network/aar_vungle	Vungle SDK	No

2.mainTemplate.gradle introduction

You must first use Unity3d to generate the mainTemplate.gradle file. As shown below:





Description for mainTemplate.gradle:

(The generated sample files are provided in the SDK. Since the gradle files generated by different versions of the Unity3d tools will be different, the developer needs to delete and regenerate the gradle files under the Unity3d tool. The SDK is for reference only.)

```
buildscript {
    repositories {
        google()
        jcenter()
    }

    dependencies {
        //Different Unity3d tools may generate different version numbers.
        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 //Additional settings are required to set when the number of lines of code exceeds 64k.
    }
    ....
}

```

3.AndroidManifest.xml introduction:

```

<?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" />

    <!--The configuration of usesCleartextTraffic is required. The main role is to make http requests available in the game (must be used)-->
    <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>

        <!--This setting is mainly for adapting to machines above 9.0 (must be used)-->
        <uses-library android:name="org.apache.http.legacy" android:required="false"/>
    </application>

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

</manifest>

```

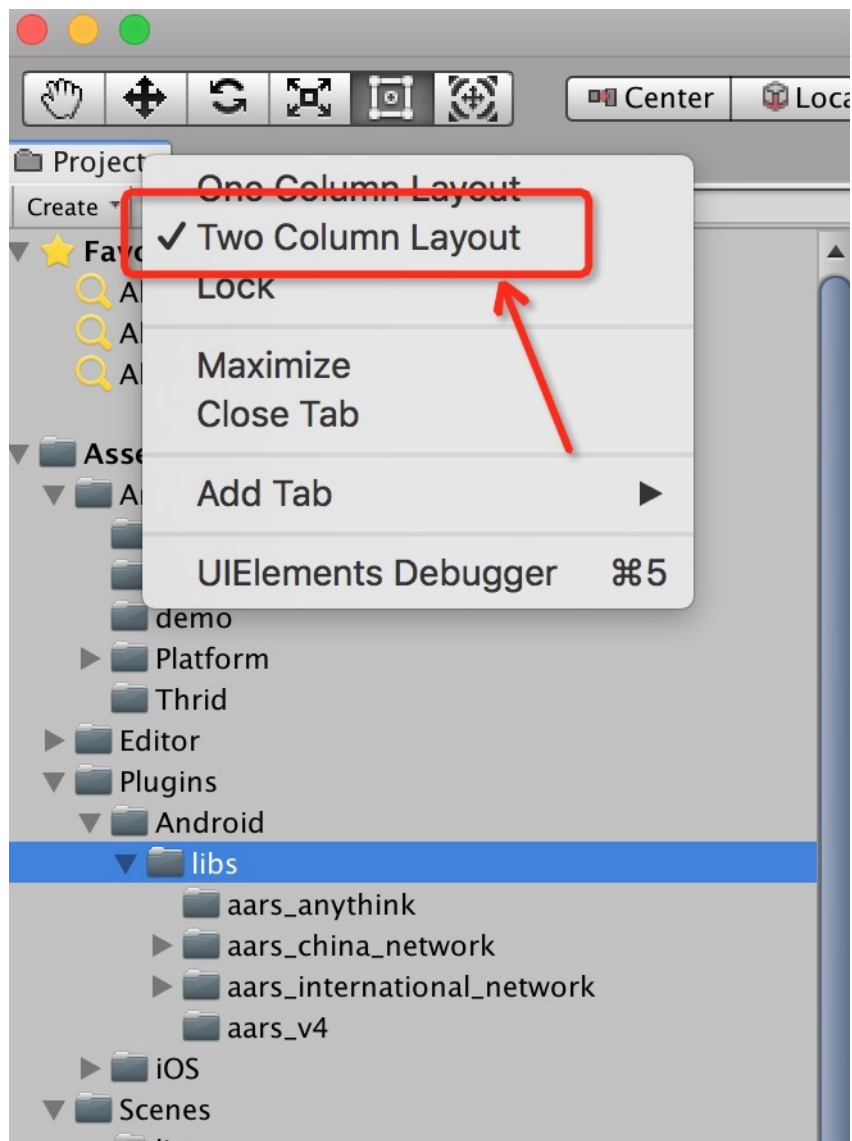
4.Instructions for merging with other third-party Android SDKs

- (1) The third-party jar and aar package must be placed in the following directory: /Assets/Plugins/Android/libs/
- (2) If the third-party sdk has resources, put the resource folder in the directory: /Assets/Plugins/Android/
- (3) If the third party has an AndroidManifest file, you need to merge it with the /Assets/Plugins/Android/AndroidManifest.xml file, Topon needs to retain the necessary parts of the above instructions.

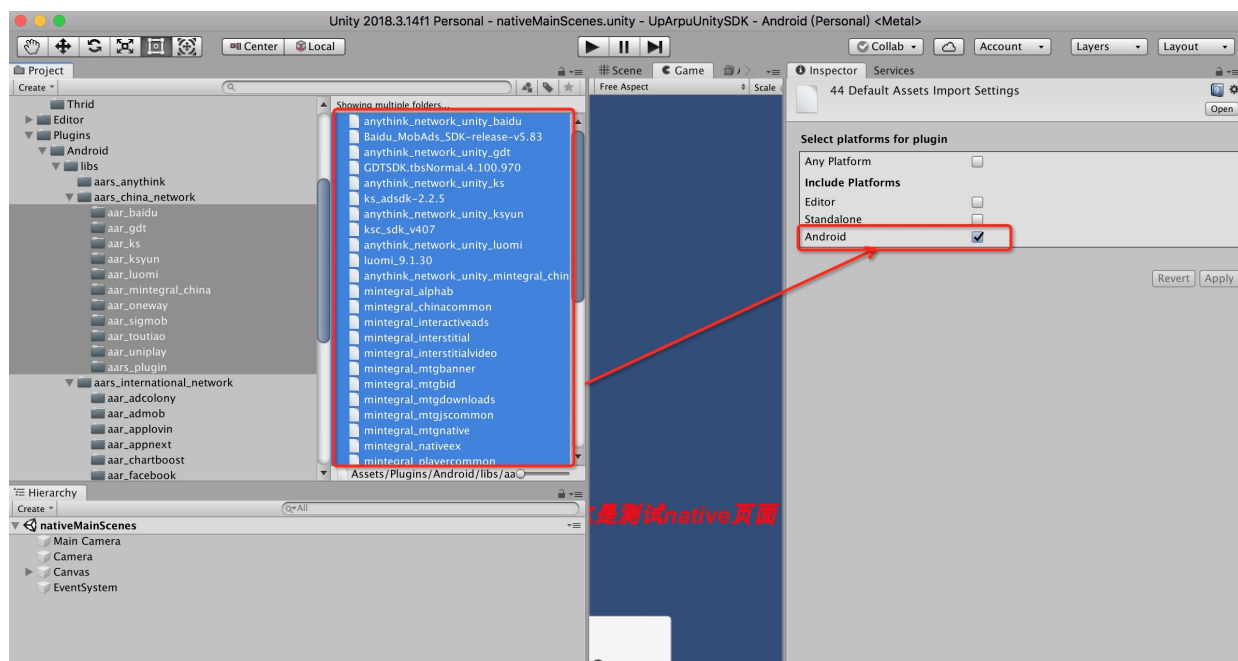
5.How to choose part of Android AAR or Jar is not packaged

Example for No packaging the Network of China :

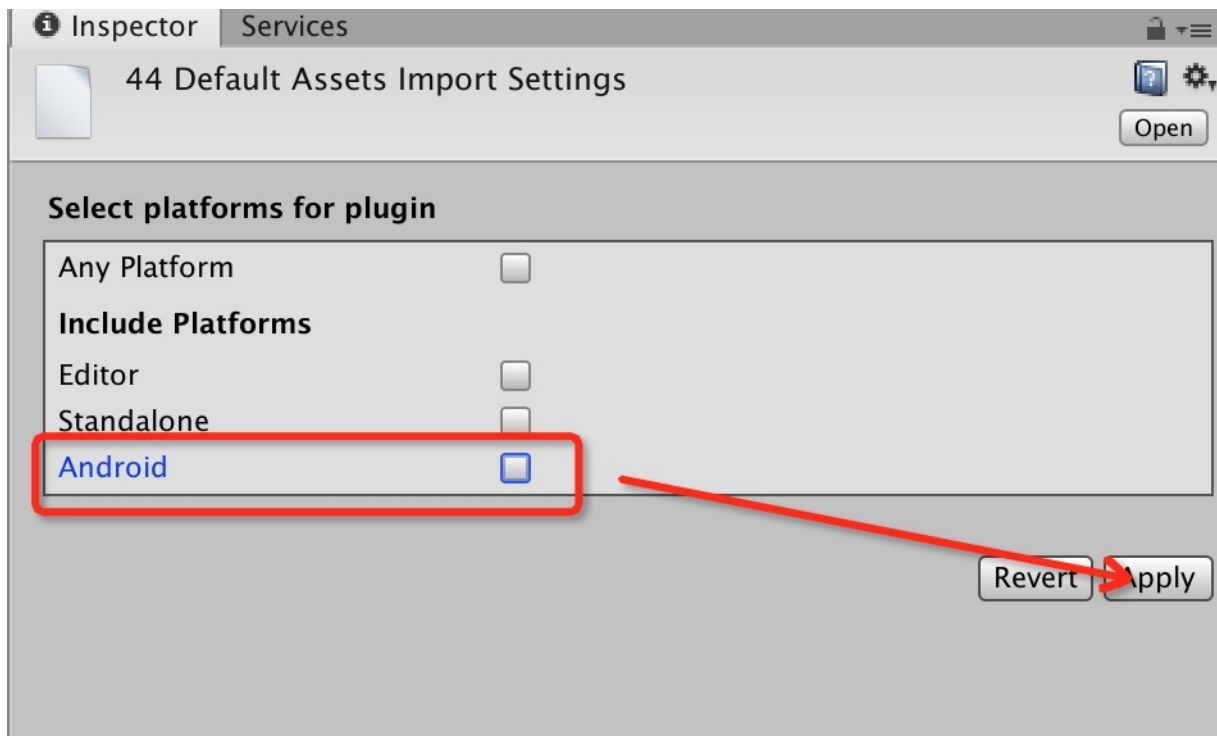
Step 1: Select a directory that shows the presentation files in Unity3d



Step 2: Select all the SDK folders in China, select all the aar packages and jar packages in the next row, and then select the platform on the far right.



Step 3: Remove the packaging options of the Android platform, and then select Apply to complete the culling package operation.



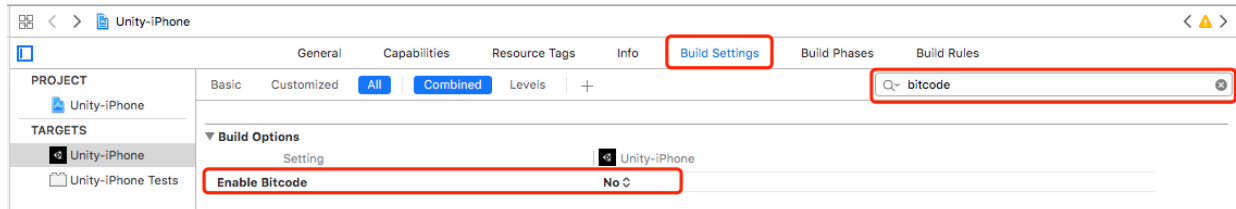
3.1.2 iOS import introduction

After compiling the Xcode project with Unity, open the Xcode project, import the required SDK according to the guidelines of each third-party platform and link the system framework and lib that it depends on. You can also see the help of each platform of Topon [Networks SDK Access Help for Topon](#)

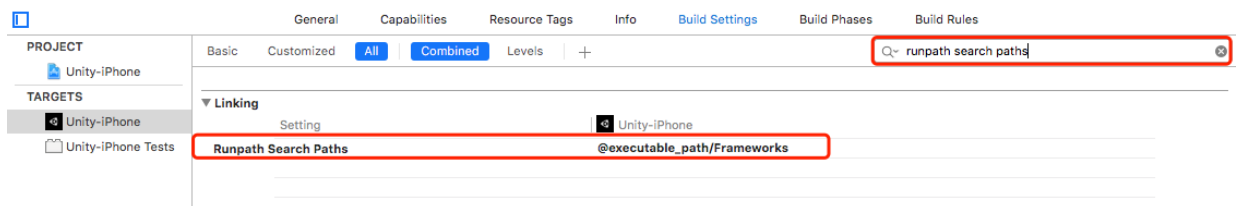
All third-party Framework packages are included in Unity's sdk package. Unneeded sdk packages can be deleted as needed. For details on which platforms need to be introduced, please see the help documentation above.

According to the information listed above, the SDK required for each third-party network is introduced and the system framework and lib are introduced according to the requirements of each SDK. The following configuration needs to be performed in the Build Settings:

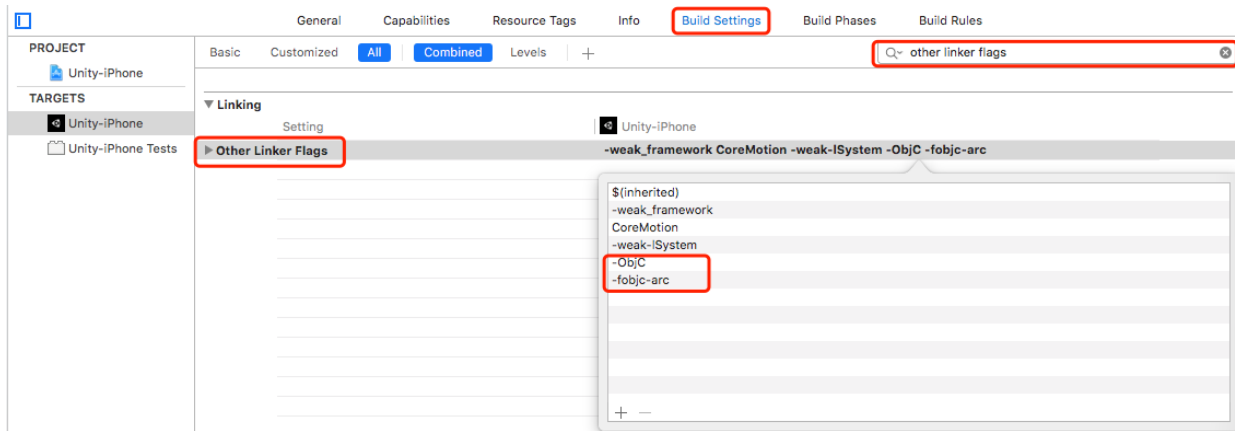
1. In the **Build Settings** of the Xcode project, search for the **bitcode** and change its value to **NO** (the current version of Unity (2018.02) compiled Xcode project, this setting defaults to Yes), as shown in Figure:



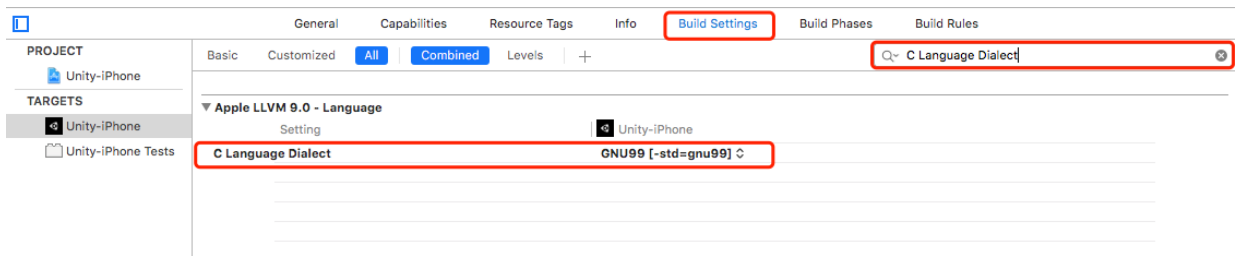
2. In the **Build Settings** of the Xcode project, search for **runpath search paths** and change its value to **@executable_path/Frameworks**, as shown in Figure:



3. In the **Build Settings** of the Xcode project, search for **other linker flags** and add **-ObjC, -fobjc-arc** based on the default values, as shown in Figure:



4. In the **Build Settings** of the Xcode project, search for **C Language Dialect** and change its value to **GNU99[-std=gnu99]**, as shown in Figure:



All the configurations described above can be programmatically made using the following **C# Editor Script**:

```
#if (UNITY_5 && UNITY_IOS) || UNITY_IPHONE
using UnityEditor.iOS.Xcode;
#endif

public static class MyBuildPostprocess
{
    [PostProcessBuild(999)]
    public static void OnPostProcessBuild(BuildTarget buildTarget, string path)
    {
        #if (UNITY_5 && UNITY_IOS) || UNITY_IPHONE
            if (buildTarget == BuildTarget.iOS)
            {
                string projectPath = path + "/Unity-iPhone.xcodeproj/project.pbxproj";

                PBXProject pbxProject = new PBXProject();
                pbxProject.ReadFromFile(projectPath);

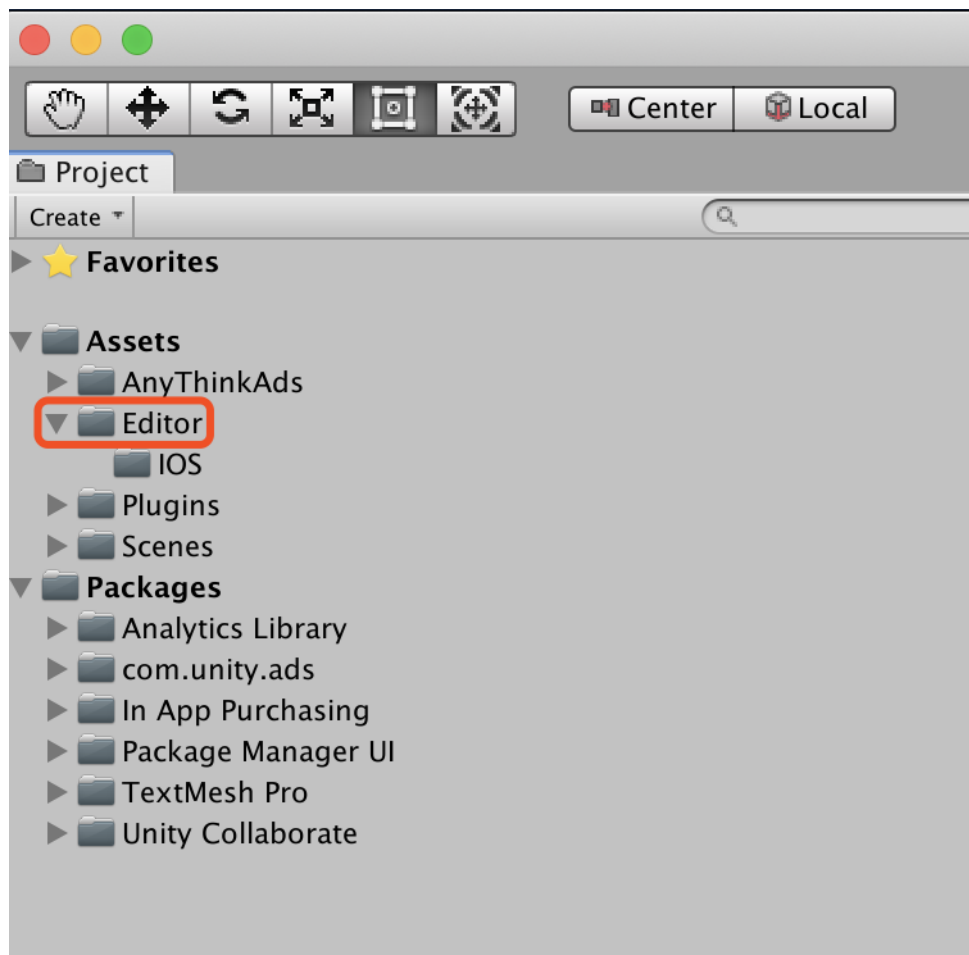
                string target = pbxProject.TargetGuidByName("Unity-iPhone");
                pbxProject.SetBuildProperty(target, "ENABLE_BITCODE", "NO");
                pbxProject.SetBuildProperty(target, "GCC_ENABLE_OBJC_EXCEPTIONS", "YES");
                pbxProject.SetBuildProperty(target, "GCC_C_LANGUAGE_STANDARD", "gnu99");

                pbxProject.AddBuildProperty(target, "OTHER_LDFLAGS", "-ObjC");
                pbxProject.AddBuildProperty(target, "OTHER_LDFLAGS", "-fobjc-arc");
                pbxProject.AddFileToBuild(target, pbxProject.AddFile("usr/lib/libxml2.tbd", "Libraries/libxml2.tbd", PBXSourceTree.Sdk));
                pbxProject.AddFileToBuild(target, pbxProject.AddFile("usr/lib/libresolv.9.tbd", "Libraries/libresolv.9.tbd", PBXSourceTree.Sdk));
                pbxProject.WriteToFile (projectPath);

                var plistPath = Path.Combine(path, "Info.plist");
                PlistDocument plist = new PlistDocument();
                plist.ReadFromFile(plistPath);
                plist.root.SetString("GADApplicationIdentifier", "ca-app-pub-9488501426181082/7319780494");
                plist.root.SetBoolean("GADIsAdManagerApp", true);
                plist.WriteToFile(plistPath);
            }
        }
    }
}

#endif
```

Note: You have to place this segment of code into a C# file under the **Editor** directory in **Unity3D IDE**:



If everything goes well, your project should compile.

4 Initialize the SDK

You can initialize **AnyThinkSDK** with the following code segment, which is extracted from our [demo project](#):

```
ATSDKAPI.setChannel("unity3d_test_channel");
ATSDKAPI.initCustomMap(new Dictionary<string, string> { { "unity3d_data", "test_data" } });
ATSDKAPI.setLogDebug(true);
ATSDKAPI.initSDK("a5c4ad280995c9", "7b4e37f819dbec652ef79c4506e14288");//Use your own app_id & app_key here
```

NOTE: There's another init method in the ATSDKAPI class that accepts a listener object that enables you to get notified on SDK initialization completion events(succeed/fail):

```
namespace AnyThinkAds.Api
{
    public class ATSDKAPI
    {
        private static readonly IATSDKAPIClient client = GetATSDKAPIClient();

        public static void initSDK(string appId, string appKey)
        {
            client.initSDK(appId, appKey);
        }

        public static void initSDK(string appId, string appKey, ATSDKInitListener listener)
        {
            client.initSDK(appId, appKey, listener);
        }
    }
}
```

However, since subsequent operations(load, show etc.) do not depend on these events, you can use the code shown above and call the loading API immediately after you've init the SDK.

5. Integrate Native Ad

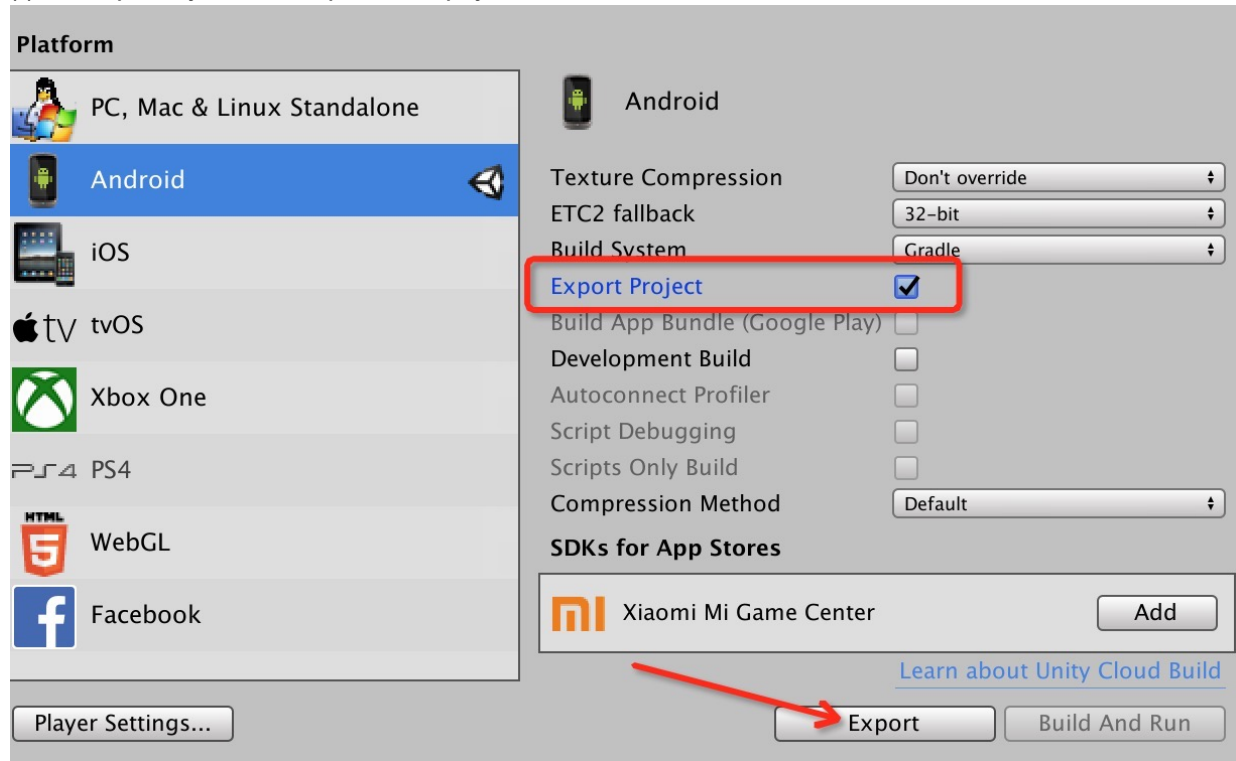
Before you go on, please make sure you've imported & initialized the SDK as described above.

Note on Android:

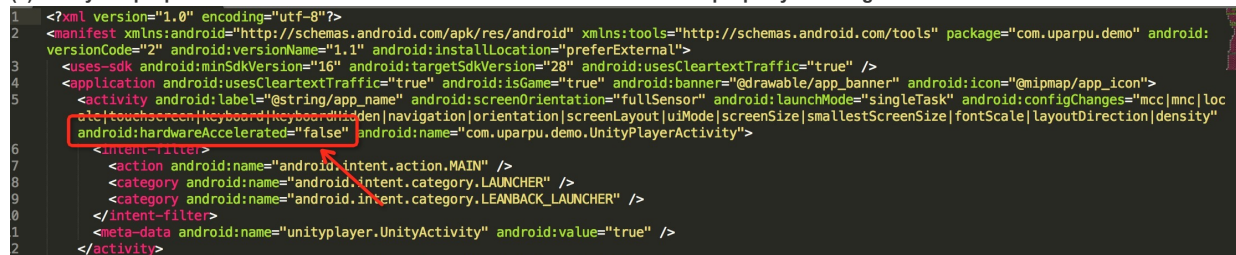
At present, it is not possible to use native video ads for Android APKs that are directly exported by Unity. Because Unity packaged APKs will turn off the game's Activity hardware acceleration by default, so videos cannot be displayed. If you need to display native advertising videos, you must use the export project method.

As follows:

(1) Select Export Project mode to export Android project



(2) Modify the properties in AndroidManifest and set the hardwareAccelerated property in the figure to true



After completing the above steps, use the current Android project to package it..

5.1 Load Native Ad

You can load native ad using the code shown below.

```
public void loadNative()
{
    Debug.Log ("Developer load native, unit id = " + mPlacementId_native_all);

    if(callbackListener == null) {
        callbackListener = new ATNativeCallbackListener();
        ATNativeAd.Instance.setListener(callbackListener);
    }

    Dictionary<string,string> gdtlocal = new Dictionary<string,string>();
    gdtlocal.Add ("gdtadtype","3");
    gdtlocal.Add ("gdtad_width","-1");
    gdtlocal.Add ("gdtad_height","-1");
    ATNativeAd.Instance.setLocalExtra(mPlacementId_native_all,gdtlocal);

    Dictionary<string,string> jsonmap = new Dictionary<string,string>();
    jsonmap.Add("age", "22");
```

```

        jsonmap.Add("sex", "lady");
        jsonmap.Add("native", "0");

        ATNativeAd.Instance.loadNativeAd(mPlacementId_native_all, jsonmap);
    }

```

NOTE: Just read on to see how you can get notified on loading success/failure events.

5.2 Show Native Ad

You can show native ad using the code shown below.

```

public void showNative()
{
    Debug.Log ("Developer show native...");
    ATNativeConfig config = new ATNativeConfig ();

    string bgcolor = "#ffffff";
    string textcolor = "#000000";
    int rootbasex = 100, rootbasey = 100;

    int x = rootbasex,y = rootbasey,width = 300*3,height = 200*3,textsize = 17;
    config.parentProperty = new ATNativeItemProperty(x,y,width,height,bgcolor,textcolor,textsize, true);

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

    //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, true);

    //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, true);

    //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, true);

    //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, true);

    //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, true);

    ATNativeAdView anyThinkNativeAdView = new ATNativeAdView(config);
    AnyThinkAds.Demo.ATManager.anyThinkNativeAdView = anyThinkNativeAdView;
    Debug.Log("Developer renderAdToScene--->");
    ATNativeAd.Instance.renderAdToScene(mPlacementId_native_all, anyThinkNativeAdView);
}

```

The trailing parameter you pass to the constructor of `ATNativeItemProperty` class denotes whether you use pixel or not; for example, on iPhone 6, if you pass 30, 120, 300, 450 for x, y, width and height respectively, the actual value pass on to Objective-C code will be 15, 60, 150, 225 while on iPhone 7 these values will be 10, 40, 100, 150; that is, the final values will be the ones you pass divide the scale of the screen of the target device.

As you can see above, we've defined a `ATNativeConfig` class for you to configure the various properties(bgColor, textColor, textSize, position, etc.) for native assets like CTA button, app icon, title text, description text, cover image and so on. Please feel free to modify the properties in the config object and see what happens according to your modifications.

If you want to remove a native ad from the screen, use the code below.

```

public void cleanView()
{
    Debug.Log ("Developer cleanView native...");
    ATNativeAd.Instance.cleanAdView(mPlacementId_native_all,AnyThinkAds.Demo.ATManager.anyThinkNativeAdView);
}

```

5.3 Implement Native Ad Listener

To get notified on the various native ad event(loading success/failure, impression and click, etc.), you can define a class that implements the **ATNativeAdListener** interface, here's an example:

```

class ATNativeCallbackListener : ATNativeAdListener
{
    public void onAdLoaded(string unitId)

```

```

    {
        Debug.Log("Developer onAdLoaded-----:" + unitId);
    }
    public void onAdLoadFail(string unitId, string code, string message)
    {
        Debug.Log("Developer onAdLoadFail-----:" + unitId + "--code:" + code + "--msg:" + message);
    }
    public void onAdImpressed(string unitId)
    {
        Debug.Log("Developer onAdImpressed-----:" + unitId);
    }
    public void onAdClicked(string unitId)
    {
        Debug.Log("Developer onAdClicked-----:" + unitId);
    }
    public void onAdVideoStart(string unitId)
    {
        Debug.Log("Developer onAdVideoStart-----:" + unitId);
    }
    public void onAdVideoEnd(string unitId)
    {
        Debug.Log("Developer onAdVideoEnd-----:" + unitId);
    }
    public void onAdVideoProgress(string unitId, int progress)
    {
        Debug.Log("Developer onAdVideoProgress-----:" + unitId);
    }
}

```

NOTE:The code segments you see in this section are extracted from **nativeScene.cs** in our [demo project](#).

6. Integrate Rewarded Video Ad

6.1 Load Rewarded Video Ad

Use the code below to load rewarded video ad:

```

public void loadVideo()
{
    if(callbackListener == null) {
        callbackListener = new ATCallbackListener();
        Debug.Log("Developer init video...unitid:" + mPlacementId_rewardvideo_all);
        ATRewardedVideo.Instance.setListener(callbackListener);
        ATRewardedVideo.Instance.addsetting(mPlacementId_rewardvideo_all, addsetting());
    }

    Dictionary<string,string> jsonmap = new Dictionary<string,string>();
    jsonmap.Add("age", "22");
    jsonmap.Add("sex", "lady");
    jsonmap.Add("rv", "1");

    ATRewardedVideo.Instance.loadVideoAd(mPlacementId_rewardvideo_all,jsonmap);
}

```

Note: See below on how to get notified on rewarded video ad events(loading sucess/failure, impression, click, video start/end and reward).

6.2 Show Rewarded Video Ad

Showing rewarded video is much more simple compared to showing native ad, juse call the showing api and pass a placement id as the argument:

```

public void showVideo()
{
    Debug.Log ("Developer show video...");
    ATRewardedVideo.Instance.showAd(mPlacementId_rewardvideo_all);
}

```

6.3 Implemente Rewarded Video Ad Listener

You can get notified on rewarded video ad events by defining a class that implemente the **ATRewardedVideoListener** interface:

```

class ATCallbackListener : ATRewardedVideoListener {
    public void onRewardedVideoAdLoaded(string unitId){
        Debug.Log("Developer onRewardedVideoAdLoaded-----");
    }

    public void onRewardedVideoAdLoadFail(string unitId, string code, string message){
        Debug.Log("Developer onRewardedVideoAdLoadFail-----:code" + code + "--message:" + message);
    }
}

```

```

    }

    public void onRewardedVideoAdPlayStart(string unitId){
        Debug.Log("Developer onRewardedVideoAdPlayStart-----");
    }

    public void onRewardedVideoAdPlayEnd(string unitId){
        Debug.Log("Developer onRewardedVideoAdPlayEnd-----");
    }

    public void onRewardedVideoAdPlayFail(string unitId, string code, string message){
        Debug.Log("Developer onRewardedVideoAdPlayFail-----code:" + code + "---message:" + message);
    }

    public void onRewardedVideoAdPlayClosed(string unitId, bool isReward){
        Debug.Log("Developer onRewardedVideoAdPlayClosed-----isReward:" + isReward);
    }

    public void onRewardedVideoAdPlayClicked(string unitId){
        Debug.Log("Developer onRewardVideoAdPlayClicked-----");
    }

    public void onReward(string unitId){
        Debug.Log("Developer onReward-----");
    }
}

```

Create an instance of this class and pass it to the listener parameter in the loading api as shown [here](#).

NOTE: The code segments in this section are extracted from **videoScenes.cs** in our [demo project](#).

7. Integrate Interstitial Ad

7.1 Load Interstitial Ad

Use the code below to load interstitial ad:

```

public void loadInterstitialAd()
{
    if(callback == null) {
        callback = new InterstitialCallback();
        ATInterstitialAd.Instance.setListener(callback);
    }

    Dictionary<string,string> jsonmap = new Dictionary<string,string>();
    jsonmap.Add("age", "22");
    jsonmap.Add("sex", "lady");
    jsonmap.Add("interstitial", "3");

    ATInterstitialAd.Instance.loadInterstitialAd(mPlacementId_interstitial_all, jsonmap);
}

```

Note: See below on how to get notified on interstitial ad events (loading success/failure, impression, click, video start/end).

7.2 Show Interstitial Ad

Almost the same as interstitial's showing api, the interstitial showing api accepts a placement id as its sole parameter:

```

public void showInterstitialAd()
{
    ATInterstitialAd.Instance.showInterstitialAd(mPlacementId_interstitial_all);
}

```

7.3 Implement Interstitial Listener

You can get notified on interstitial ad events by defining a class that implements the **ATInterstitialAdListener** interface:

```

class InterstitialCallback : ATInterstitialAdListener
{
    public void onInterstitialAdClick(string unitId)
    {
        Debug.Log("Developer callback onInterstitialAdClick :" + unitId);
    }

    public void onInterstitialAdClose(string unitId)
    {
        Debug.Log("Developer callback onInterstitialAdClose :" + unitId);
    }
}

```

```

    }

    public void onInterstitialAdEndPlayingVideo(string unitId)
    {
        Debug.Log("Developer callback onInterstitialAdEndPlayingVideo :" + unitId);
    }

    public void onInterstitialAdFailedToPlayVideo(string unitId, string code, string message)
    {
        Debug.Log("Developer callback onInterstitialAdFailedToPlayVideo :" + unitId + "--code:" + code + "--msg:" + message);
    }

    public void onInterstitialAdLoad(string unitId)
    {
        Debug.Log("Developer callback onInterstitialAdLoad :" + unitId);
    }

    public void onInterstitialAdLoadFail(string unitId, string code, string message)
    {
        Debug.Log("Developer callback onInterstitialAdLoadFail :" + unitId + "--code:" + code + "--msg:" + message);
    }

    public void onInterstitialAdShow(string unitId)
    {
        Debug.Log("Developer callback onInterstitialAdShow :" + unitId);
    }

    public void onInterstitialAdStartPlayingVideo(string unitId)
    {
        Debug.Log("Developer callback onInterstitialAdStartPlayingVideo :" + unitId);
    }

    public void onInterstitialAdFailedToShow(string unitId)
    {
        Debug.Log("Developer callback onInterstitialAdFailedToShow :" + unitId);
    }
}

```

NOTE:The code segments you see in this section are extracted from **interstitialScenes.cs** in our demo project.

8. Integrate Banner Ad

8.1 Load Banner Ad

Use the code below to load banner ad:

```

public void loadBannerAd()
{
    if(bannerCallback == null) {
        bannerCallback = new BannerCallback();
        ATBannerAd.Instance.setListener(bannerCallback);
    }

    Dictionary<string, object> jsonmap = new Dictionary<string,object>();
    jsonmap.Add("age", "22");
    jsonmap.Add("sex", "lady");
    jsonmap.Add("banner", "2");
    ATSize bannerSize = new ATSize(this.screenWidth, 100, true);
    jsonmap.Add(ATBannerAdLoadingExtra.kATBannerAdLoadingExtraBannerAdSizeStruct, bannerSize);
    ATBannerAd.Instance.loadBannerAd(mPlacementId_native_all, jsonmap);
}

```

Read on to see how to get notified on banner ad events like loading success/failure, impression and click.

8.2 Show Banner Ad

You can show banner ad using the following code:

```

public void showBannerAd()
{
    ATRect arpuRect = new ATRect(0,70, screenWidth, 100, true);
    ATBannerAd.Instance.showBannerAd(mPlacementId_native_all, arpuRect);
}

```

The trailing parameter you pass to the constructor of `ATRect` class denotes whether you use pixel or not; for example, on iPhone 6, if you pass 30, 120, 300, 450 for x, y, width and height respectively, the actual value pass on to Objective-C code will be 15, 60, 150, 225 while on iPhone 7 these values will be 10, 40, 100, 150; that is, the final values will be the ones you pass divide the scale of the screen of the target device.

Remove a banner from the screen with the code below if you want to:

```
public void removeBannerAd()
{
    ATBannerAd.Instance.cleanBannerAd(mPlacementId_native_all);
}
```

If you only want to **temporarily hide** a banner (instead of **permanently removing** it from the screen), use the code here:

```
public void hideBannerAd()
{
    ATBannerAd.Instance.hideBannerAd(mPlacementId_native_all);
}
```

After you hide a banner, you can reshow it with the code below:

```
public void reshowBannerAd()
{
    ATBannerAd.Instance.showBannerAd(mPlacementId_native_all);
}
```

NOTE: Please note here that the showBannerAd method here does not accept a rect parameter, which is different from when you first show a banner.

The difference between removing a banner and hiding a banner is that removing a banner from the screen also destroys it (meaning that before you show it again you have to load it first) while you can reshow a previously hidden banner by just calling showBannerAd method **without passing the ATRect parameter**.

8.3 Implement Banner Ad Listener

To get notified on the various banner ad events (loading success/failure, impression&click), just define a class that implement the **ATBannerAdListener** interface:

```
class BannerCallback : ATBannerAdListener
{
    public void onAdAutoRefresh(string unitId)
    {
        Debug.Log("Developer callback onAdAutoRefresh : " + unitId);
    }

    public void onAdAutoRefreshFail(string unitId, string code, string message)
    {
        Debug.Log("Developer callback onAdAutoRefreshFail : " + unitId + "--code:" + code + "--msg:" + message);
    }

    public void onAdClick(string unitId)
    {
        Debug.Log("Developer callback onAdClick : " + unitId);
    }

    public void onAdClose(string unitId)
    {
        Debug.Log("Developer callback onAdClose : " + unitId);
    }

    public void onAdImpress(string unitId)
    {
        Debug.Log("Developer callback onAdImpress : " + unitId);
    }

    public void onAdLoad(string unitId)
    {
        Debug.Log("Developer callback onAdLoad : " + unitId);
    }

    public void onAdLoadFail(string unitId, string code, string message)
    {
        Debug.Log("Developer callback onAdLoadFail : " + unitId + "--code:" + code + "--msg:" + message);
    }
}
```

NOTE: The code segments you see in this section are extracted from **bannerScene.cs** in our [demo project](#).

9. On Splash Ad

It's highly recommended that splash ad should be integrated directly using native API (Objective-C/Java) in Xcode/Android Studio instead of C# script in Unity3D. The reason for doing this is that splash ad should be loaded&shown as soon as the app finishes launching. If you integrate splash in Unity3D

using C#, it can't load&show ad until Unity game engine has been launched, which happens after the app finishes launch and takes much more time. For more information on how to integrate splash using Objective-C/Java, please refer to [iOS Integration Guide](#) & [Android Integration Guide](#).