

ALIBABA CLOUD

# 阿里云

音视频通信  
Demo

文档版本：20210226

 阿里云

## 法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

# 通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
<b>粗体</b>	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击 <b>确定</b> 。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[ ] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

# 目录

1.Demo体验	05
2.Android Demo	08
3.iOS Demo	12
4.Mac Demo	16
5.Windows Demo	20
6.Web Demo	23
7.Unity Demo	26

# 1.Demo体验

您可以通过阅读本文，了解RTC Demo下载地址，快速体验Demo。

## Demo体验

您可以通过移动端使用钉钉扫描下方二维码，下载App进行体验。

**说明** Web体验，Windows端目前仅支持Chrome浏览器。Mac端支持Chrome浏览器和Safari浏览器。

当您遇到RTC不支持的情况，请查看是否插入音频设备（麦克风和摄像头），查看浏览器和系统是否禁用音视频。

iOS	Android	Mac OS	Windows	Web
		敬请期待	敬请期待	音视频通话
				互动大班课
				超级小班课

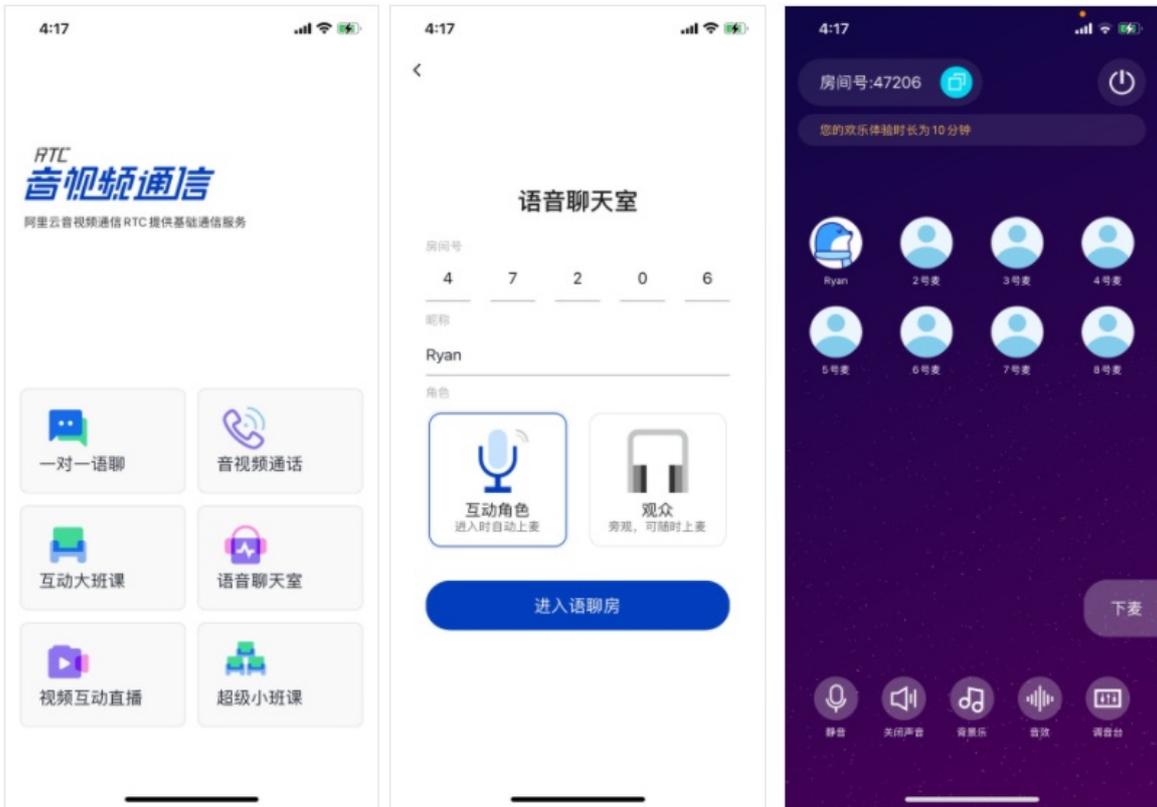
## Demo源码下载

场景	平台	场景简介	示例代码	版本更新
1对1语音聊天	iOS、Android	<a href="#">1对1语音聊天简介</a>	<a href="#">源码下载</a>	无
音视频通话	iOS、Android、Web	<a href="#">音视频通话简介</a>	<a href="#">源码下载</a>	无
互动大班课	<ul style="list-style-type: none"> <li>学生端：iOS、Android</li> <li>教师端：Web</li> </ul>	<a href="#">互动大班课简介</a>	<a href="#">源码下载</a>	无
语音聊天室	iOS、Android	<a href="#">语音聊天室简介</a>	<a href="#">源码下载</a>	无
视频互动直播	iOS、Android	<a href="#">视频互动直播简介</a>	<a href="#">源码下载</a>	无
超级小班课	<ul style="list-style-type: none"> <li>学生端：iOS、Android</li> <li>教师端：Web</li> </ul>	<a href="#">超级小班课简介</a>	<a href="#">源码下载</a>	无

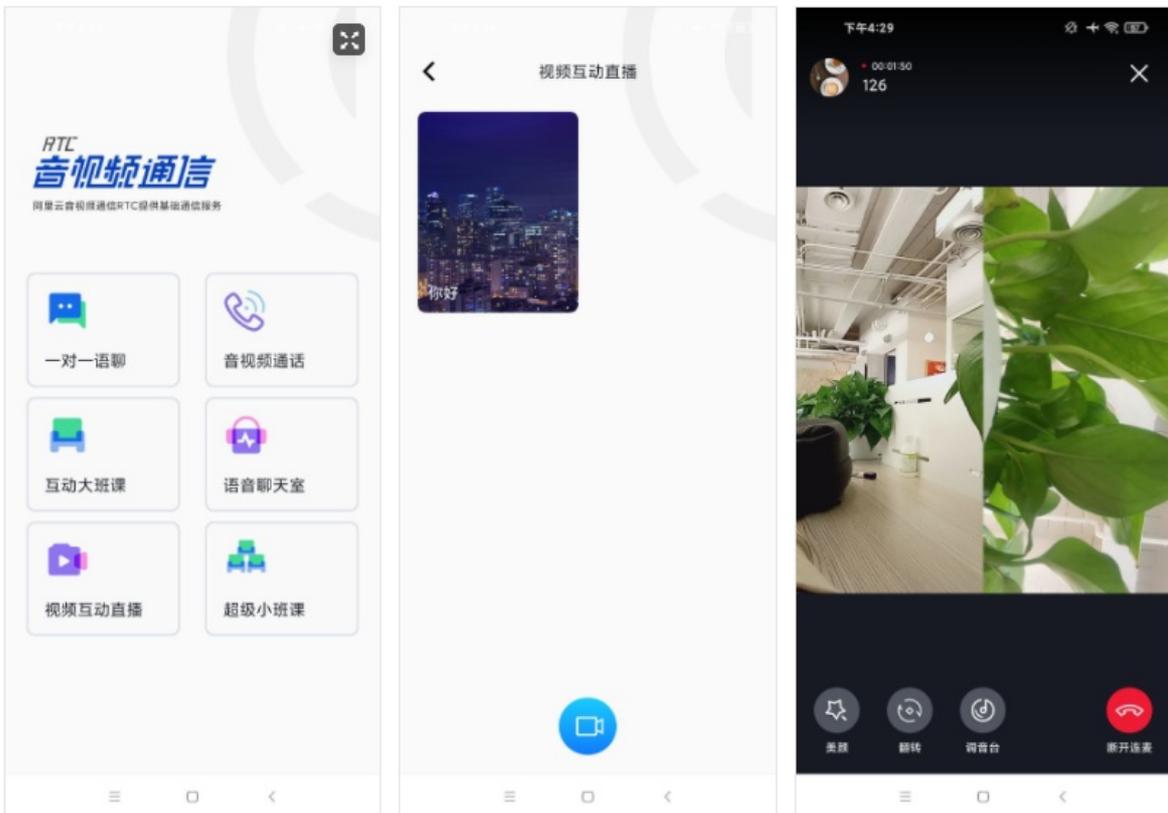
## 界面效果展示

RTC产品体验Demo的效果图如下所示：

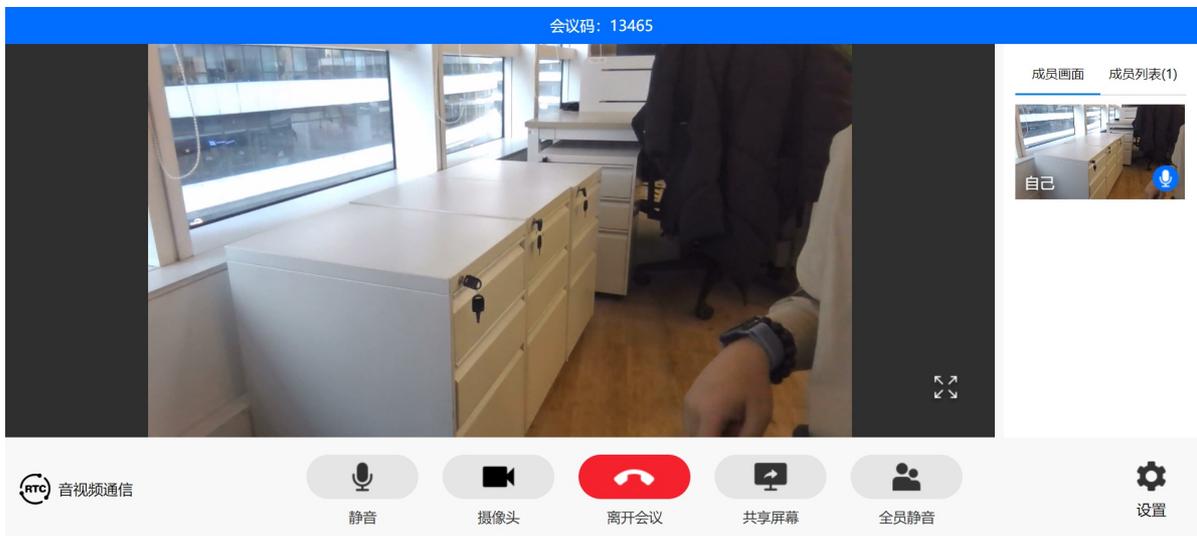
- iOS端——语音聊天室



• Android端——视频互动直播



• Web端



## 2.Android Demo

您可以阅读本文，了解快速运行Android Demo的操作方法，实现加入频道和远端用户进行音视频通信。

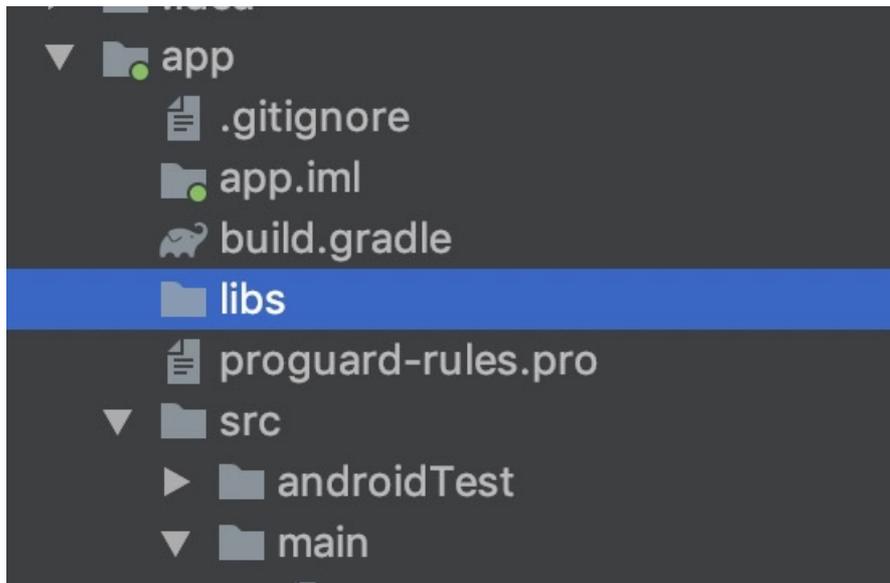
### 前提条件

在执行Demo步骤之前，您需要从控制台获取鉴权信息，具体操作请参见[生成Token](#)。

您需要下载示例代码，详情请参见[SDK下载](#)。

### 操作步骤

1. 在该目录下的libs文件夹中添加RTC SDK，如果没有libs文件夹，需要手动创建。



2. 打开AliRtcChatActivity.java文件，配置鉴权信息参数。

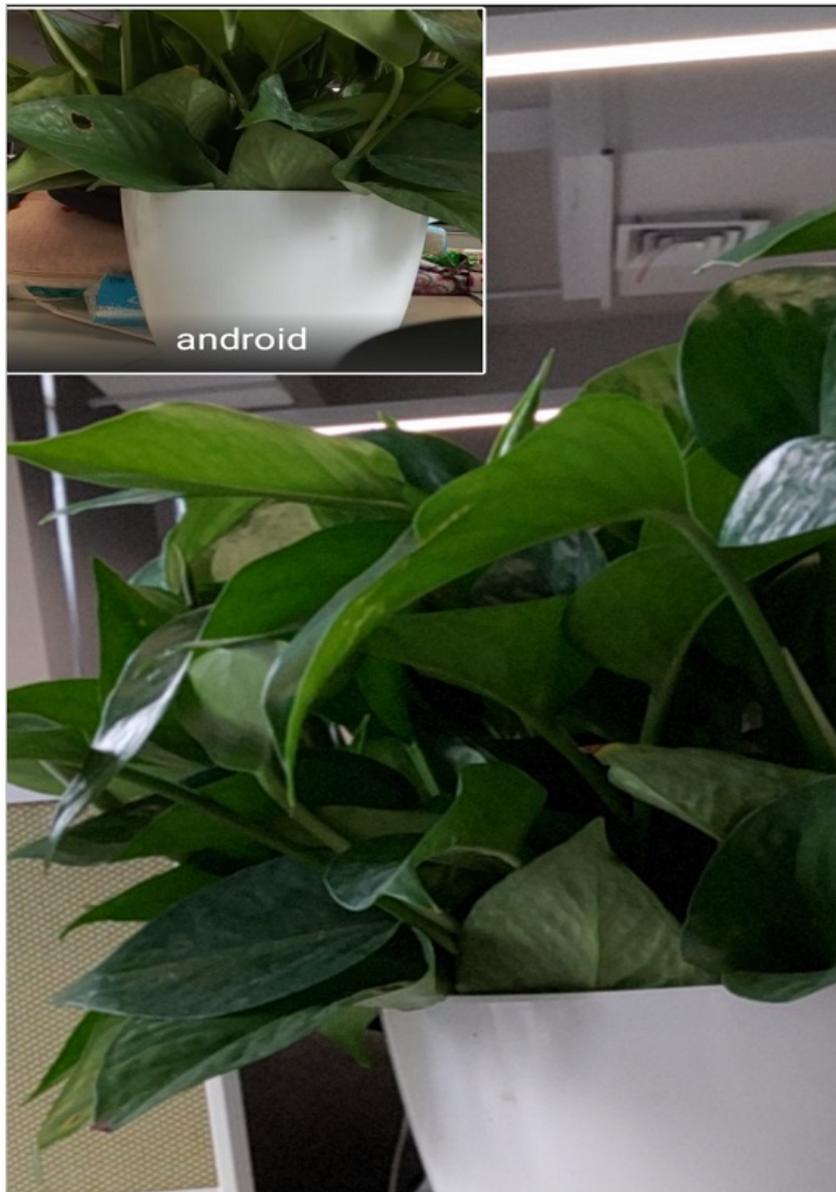
```
private void joinChannel() {
    if (mAliRtcEngine == null) {
        return;
    }
    AliRtcAuthInfo userInfo = new AliRtcAuthInfo();
    userInfo.setAppid("zwdm****");
    userInfo.setNonce("AK-d8847d08-c8b3-4800-84e3-7f6c4d65f96d");
    userInfo.setGslb(new String[]{"https://rgslb.rtc.aliyuncs.com"});
    userInfo.setTimestamp(1589379613);
    userInfo.setToken("e48d39484c91a26****");
    userInfo.setConferenceld("1234");
    userInfo.setUserId("testId");
    /*
     *设置自动发布和订阅，只能在加入频道之前设置。
     *autoPub：是否自动发布，取值true|false。
     *autoSub：是否自动订阅，取值true|false。
     */
    mAliRtcEngine.setAutoPublishSubscribe(true, true);
    // 加入频道，需要填写鉴权信息和用户名。
    mAliRtcEngine.joinChannel(userInfo,"用户名");
}
```

3. 运行Demo。Demo运行成功后进入音视频通话界面，您可以看到本地已经开启的预览视图，然后会自动加入频道，等待远端用户的加入。

 **注意** Android端Demo必须在Android 4.4及以上系统的真机上运行。



如果该频道中有其他用户即可开始实时音视频通话。



## 3.iOS Demo

您可以阅读本文，了解快速运行iOS Demo的操作方法，实现加入频道和远端用户进行音视频通信。

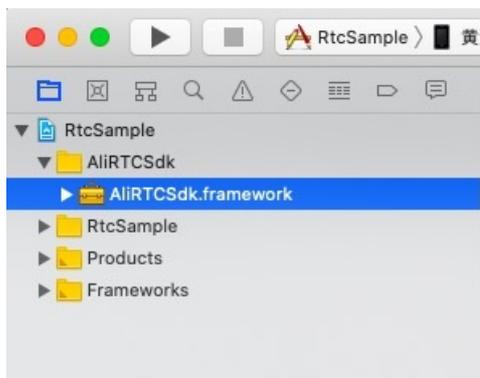
### 前提条件

在执行Demo步骤之前，您需要从控制台获取鉴权信息，具体操作请参见[生成Token](#)。

您需要下载示例代码，详情请参见[SDK下载](#)。

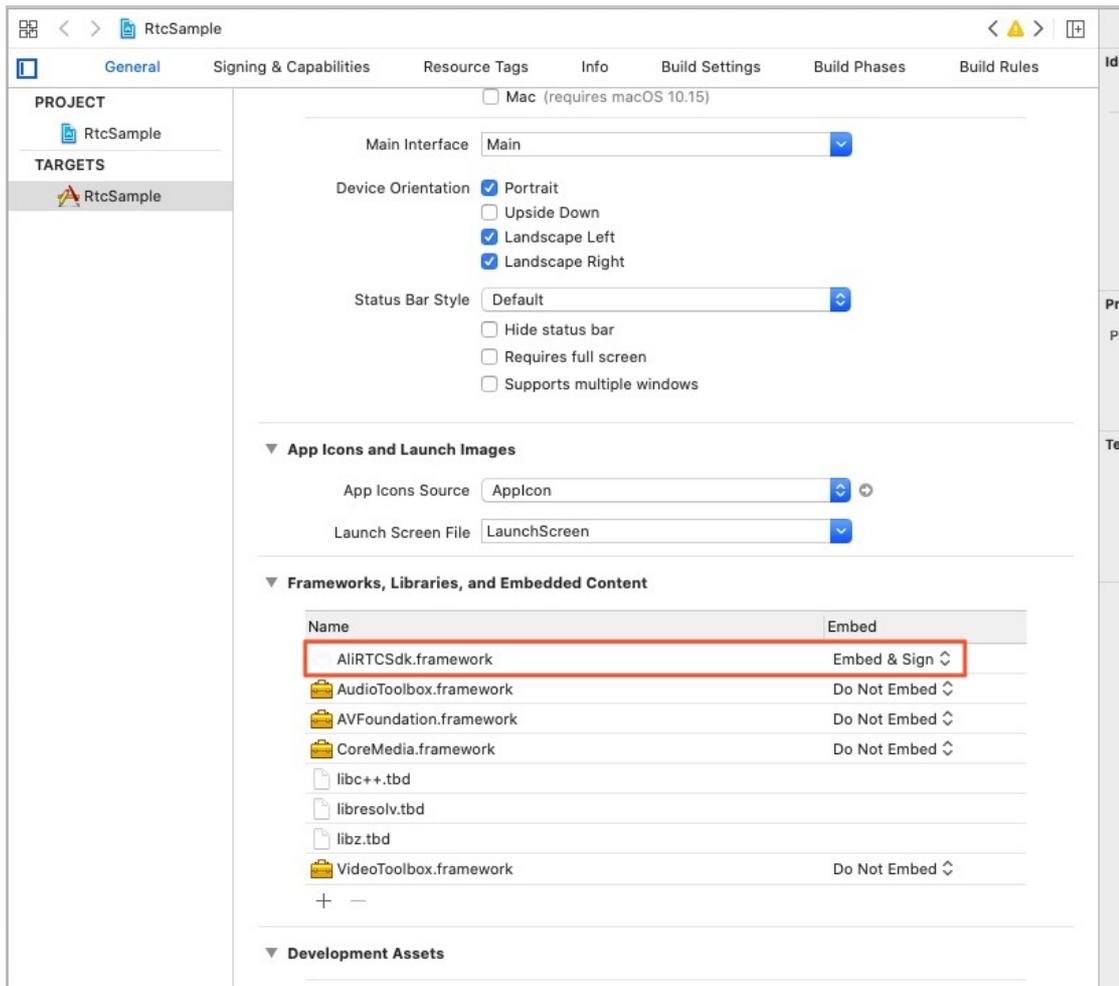
### 操作步骤

1. 下载SDK，复制一份到该文件夹下。

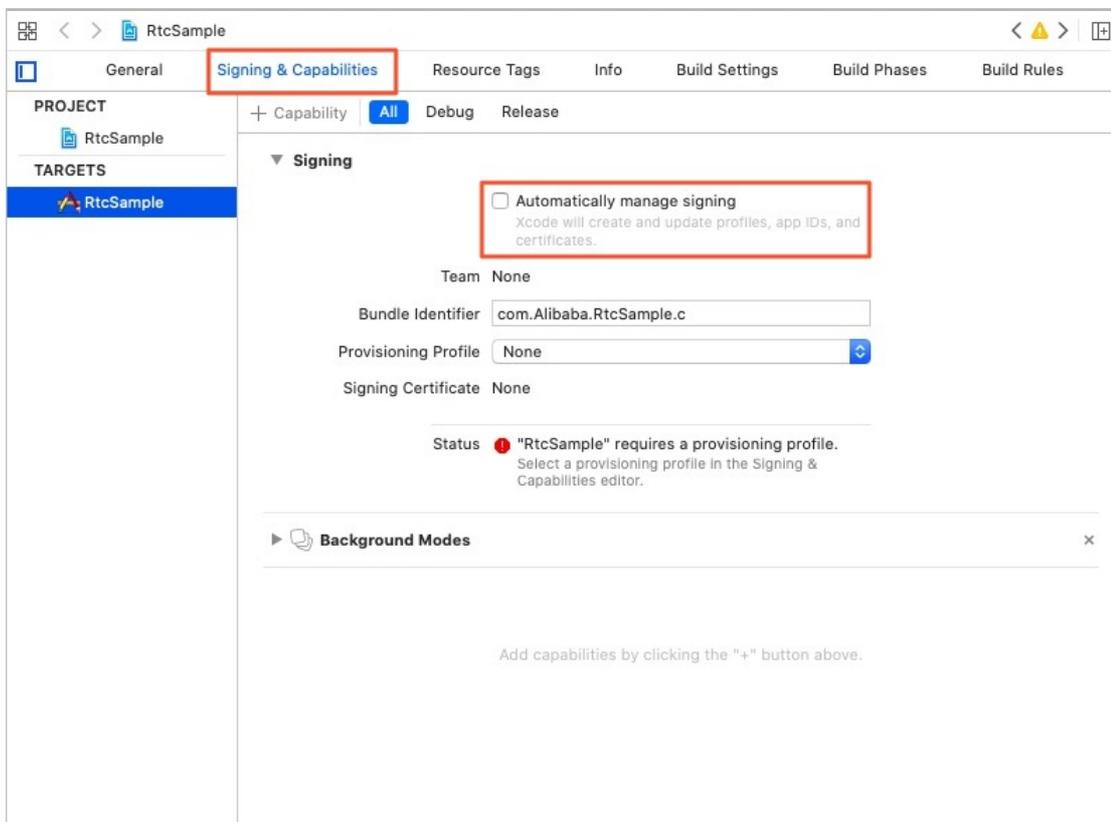


2. 配置参数。

i. 将AliRTCsdk.framework设置动态库，如下图所示：



- ii. 证书配置Demo工程需要真机环境，因此需要选择自己的证书和profile文件，修改Bundle Identifier为与自己证书匹配的值。Xcode9.0之后版本可以通过选中Automatically manage signing选项由Xcode自动管理。



- iii. 打开RtCSampleChatViewController.m文件，配置鉴权信息AliRtcAuthInfo（在joinBegin方法中）。

```

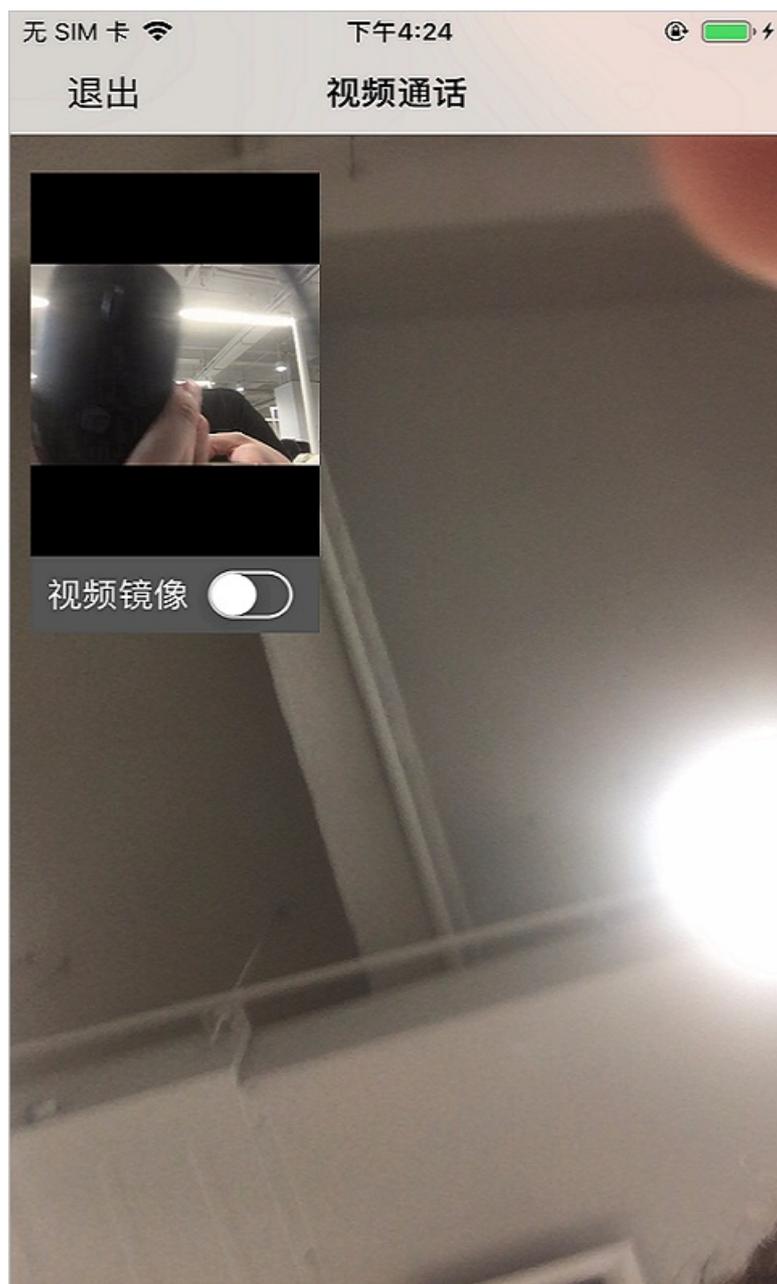
/*
  NSString *AppID = @"aoe****";
  NSString *userID = @"f303d59acae073****";
  NSString *channelID = @"23****";
  NSString *nonce = @"AK-7c8f947b-e55e-4ed5-ab09-bc849****";
  long long timestamp = 1585119606; //过期时间戳，例如：1560588594代表过期时间为2019-06-15 16:49:54。
  NSString *token = @"70f77b8****";
  NSArray <NSString *> *GSLB = @[@"https://rgslb.rtc.aliyuncs.com"];
  NSArray <NSString *> *agent = @[@""];
*/

```

3. 运行。

Demo运行成功进入首页，您可以看到本地预览视图并进入房间。如果该频道中有其他用户即可开始实时音视频通话。

**注意** iOS端Demo必须在真机上运行，设备支持iPhone5及以上，系统iOS9及以上版本。



## 4.Mac Demo

您可以阅读本文，了解快速运行Mac Demo的操作方法，实现加入频道和远端用户进行音视频通信。

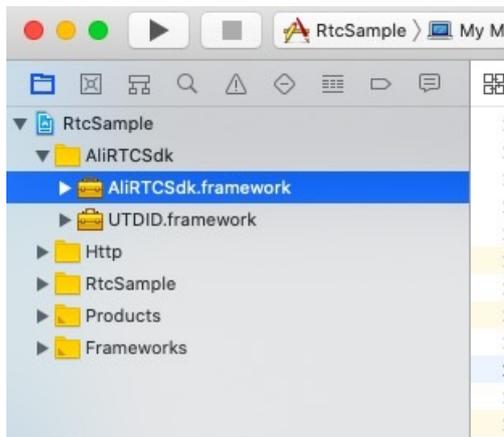
### 前提条件

在执行Demo步骤之前，您需要从控制台获取鉴权信息，具体操作请参见[生成Token](#)。

您需要下载示例代码，详情请参见[SDK下载](#)。

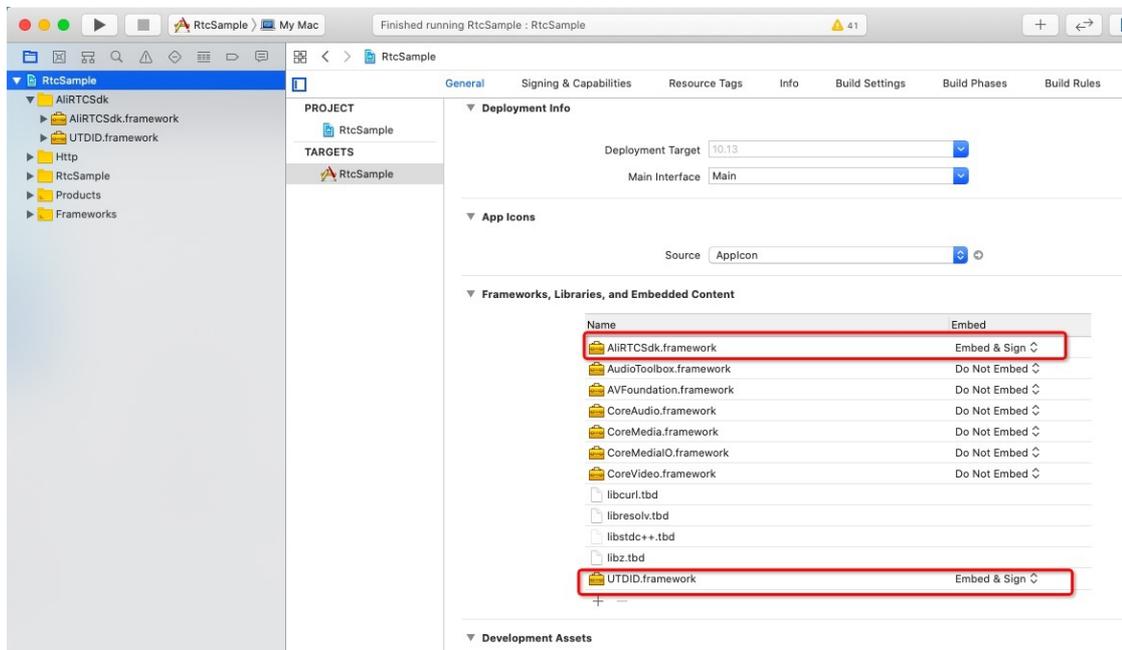
### 操作步骤

1. 下载SDK，复制一份到该文件夹下。



2. 配置参数。

- i. 将AliRTCSdk.framework和UTDID.framework设置动态库，如下图所示：



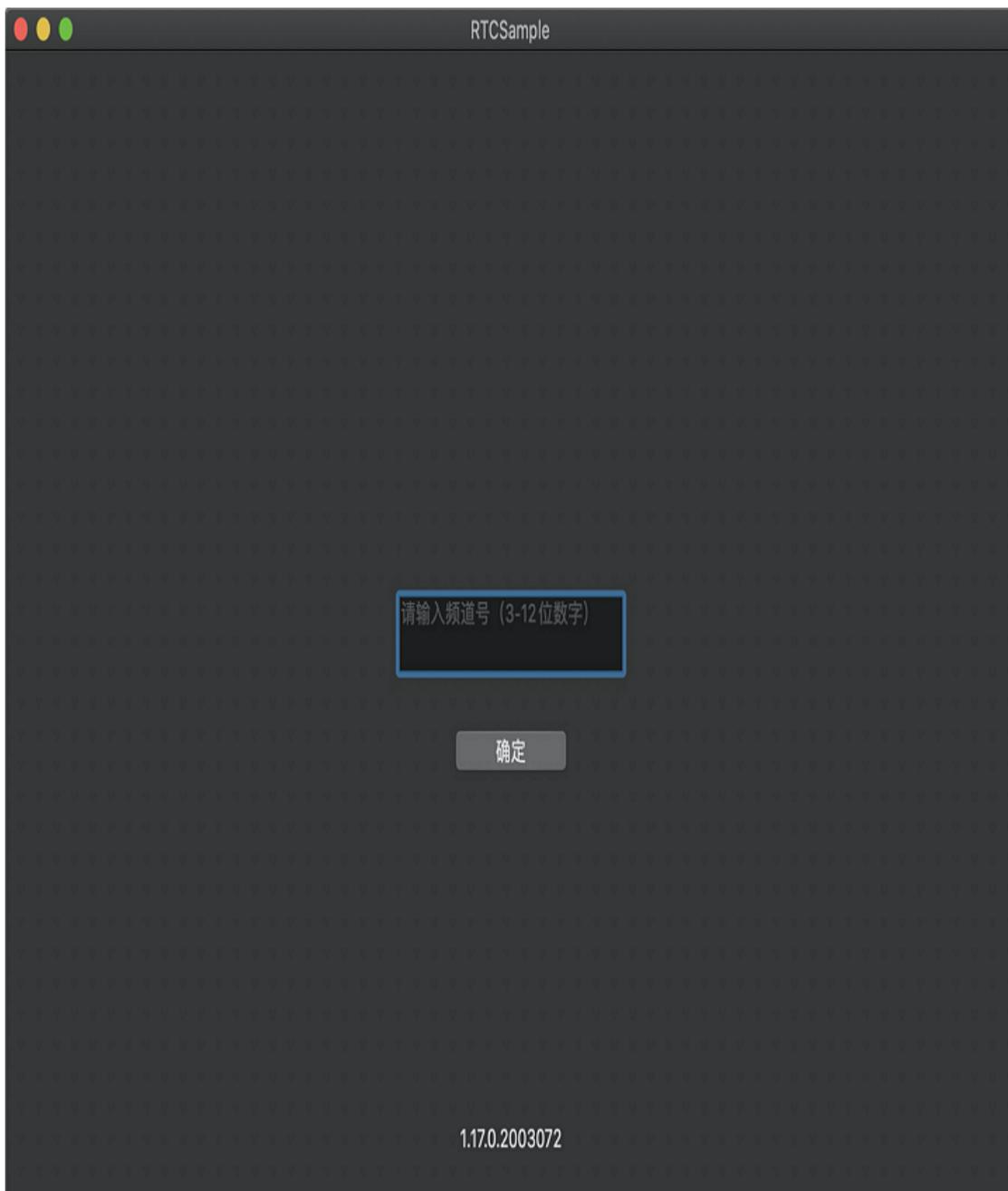
- ii. 打开 `ViewController.m` 文件，配置鉴权信息 `AliRtcAuthInfo`（在 `sureButtonClick` 方法中）。

```
/*
    NSString *AppID = @"aoe****";
    NSString *userID = @"f303d****";
    NSString *channelID = @"23****";
    NSString *nonce = @"AK-7c8f947b-e55e-4ed5-ab09-bc849efe5bee";
    long long timestamp = 1585119606;
    NSString *token = @"70f77b873****";
    NSArray<NSString*> *GSLB = @[@"https://rgslb.rtc.aliyuncs.com"];
    NSArray<NSString*> *agent = @[@""];
*/
```

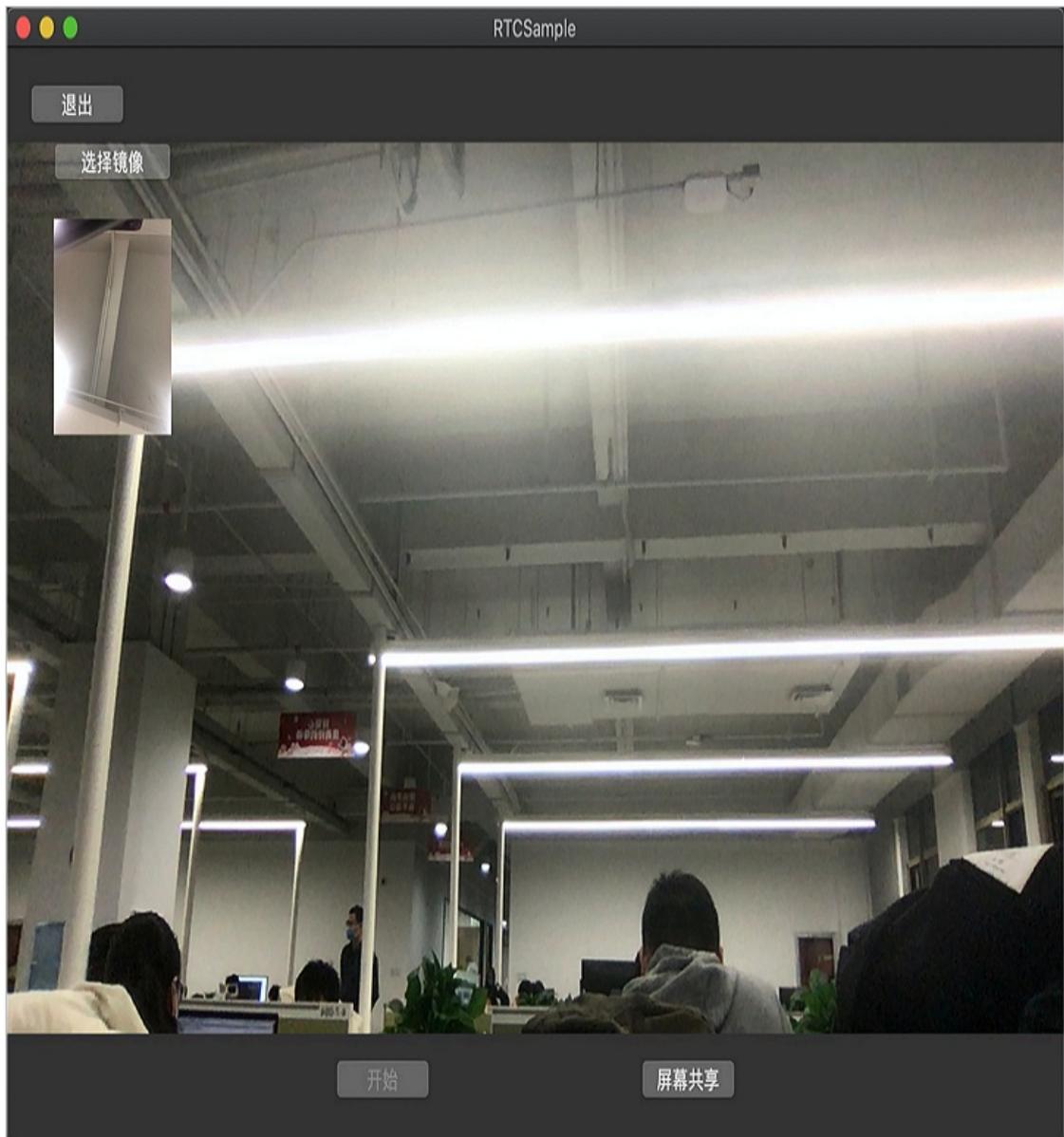
3. 运行。

 说明 Mac端Demo如果使用Mac mini等不包含自带摄像头和麦克风的设备，需要插入外置摄像头和麦克风。系统版本支持macOS 10.12及以上。

- i. Demo运行成功进入首页，输入频道号并单击确定，进入下一页面。



- ii. 进入频道页面后，可以看到本地已经开启的预览视图。单击开始加入频道，如果该频道中有其他用户即可开始实时音视频通话。



您也可以单击屏幕共享把自己的屏幕分享给远端用户。

# 5.Windows Demo

通过本文，您可以了解快速运行Windows Demo的操作方法，实现加入频道和远端用户进行音视频通信。

## 前提条件

在执行Demo步骤之前，您需要从控制台获取鉴权信息，具体操作请参见[生成Token](#)。

您需要下载示例代码，详情请参见[SDK下载](#)。

## 操作步骤

1. 下载 *MFC Demo* 文件夹和最新版本 SDK 压缩包。
2. 将 SDK 解压到 *MFC Demo* 文件夹中。
3. 打开 Demo 中的 *RTCSampleMainInterface.cpp* 文件并定位到第 1107 行代码，配置鉴权信息 `AuthInfo`。

```
authinfo.channel = CStringToAliString(m_sLoginInfo.s_strRoomID);
authinfo.user_id = "a9e28****";
authinfo.appid = "aoe****";
authinfo.nonce = "AK-e5babe72e81a54fb9f580d4****";
authinfo.token = "39cf7cde****";
authinfo.gslb.AddString("https://rgslb.rtc.aliyuncs.com");
authinfo.agent.AddString("");
authinfo.timestamp = 1585363086;//过期时间戳，例如：1560588594代表过期时间为2019-06-15 16:49:54。
```

```
authinfo.channel = CStringToAliString(m_sLoginInfo.s_strRoomID);
authinfo.user_id = "a9e28****";
authinfo.appid = "aoe****";
authinfo.nonce = "AK-e5babe72e81a54fb9f580d4****";
authinfo.token = "39cf7cde****";
authinfo.gslb.AddString("https://rgslb.rtc.aliyuncs.com");
authinfo.agent.AddString("");
authinfo.timestamp = 1585363086;//过期时间戳，例如：1560588594代表过期时间为2019-06-15 16:49:54。
```

4. 运行 Demo。

- i. Demo运行成功进入首页，单击设置进入参数设置界面，服务地址您可以不填。

② 说明 Windows端Demo如果使用不包含自带摄像头和麦克风的设备，需要插入外置摄像头和麦克风。系统支持Windows XP (SP3)、Windows 7、Windows 8.X、Windows 10，不支持Windows XP (SP3) 以下版本。

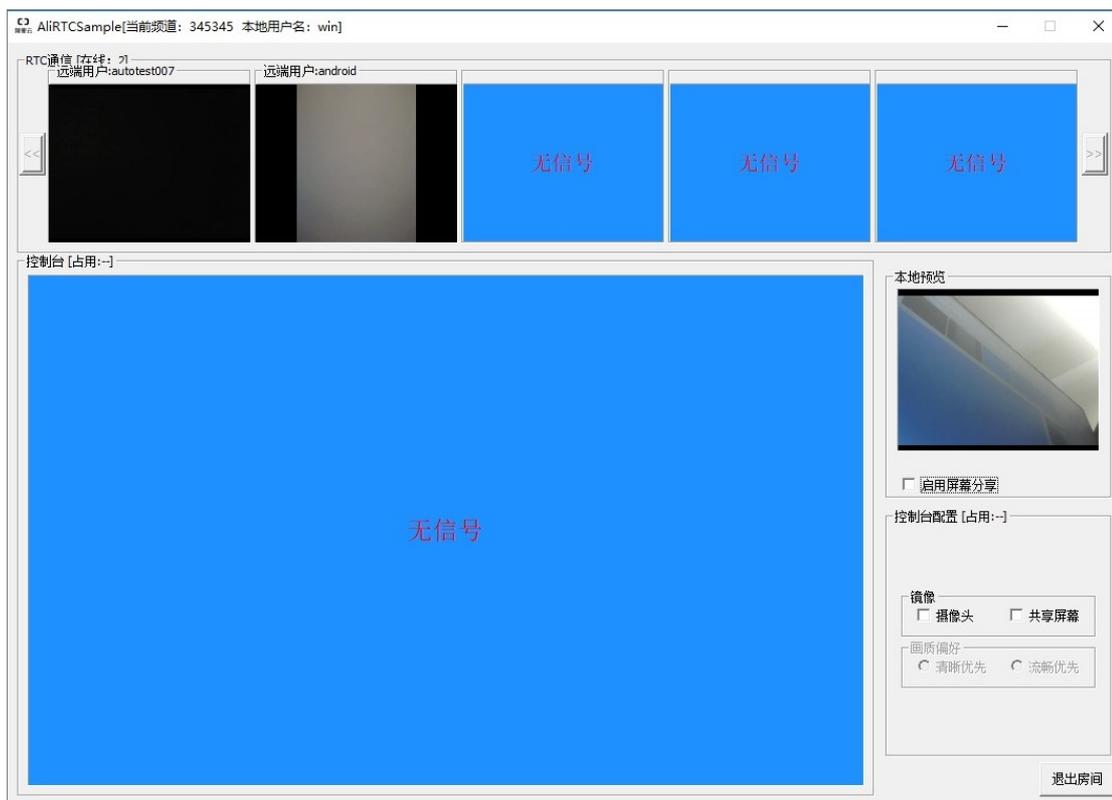


如果您想和Web端连接，请选中**兼容H5模式**。推送视频流请选中**视频流**，推送音频流请选中**音频流**。

- ii. 关闭设置页面，在首页输入相应**频道号**和**用户名**，单击**创建/加入房间**按钮，进入下一页面。



- iii. 进入频道页面后，可以看到本地已经开启的预览视图。单击开始加入频道，如果该频道中有其他用户，即可开始实时音视频通话。



您也可以勾选**屏幕共享**把自己的屏幕分享给远端用户。

## 6.Web Demo

本文为您介绍运行Web Demo的前提条件及具体步骤。在您成功运行Demo后，您可以进入房间和远端用户进行音视频通信。

### 前提条件

在执行Demo步骤之前，您需要从控制台获取鉴权信息，具体操作请参见[生成Token](#)。

您需要下载示例代码，详情请参见[SDK下载](#)。

### 操作步骤

1. 打开`index.html`文件。

- 集成SDK。

下载最新Web SDK添加到代码同层目录，并打开相关注释。

```
<!-- 请从官网下载最新sdk, 添加到同层目录, 打开下方注释 -->
<!-- <script src="./aliyun-webrtc-sdk.js"></script> -->
```

- 定位到314行代码处并配置 `AppId`和 `AppKey`。

**说明** 从控制台获取的Token仅为开发测试使用，正式上线有被攻击风险。建议您自己搭建服务端生成Token，并使用Https协议。搭建服务端请参见[服务端生成Token](#)。

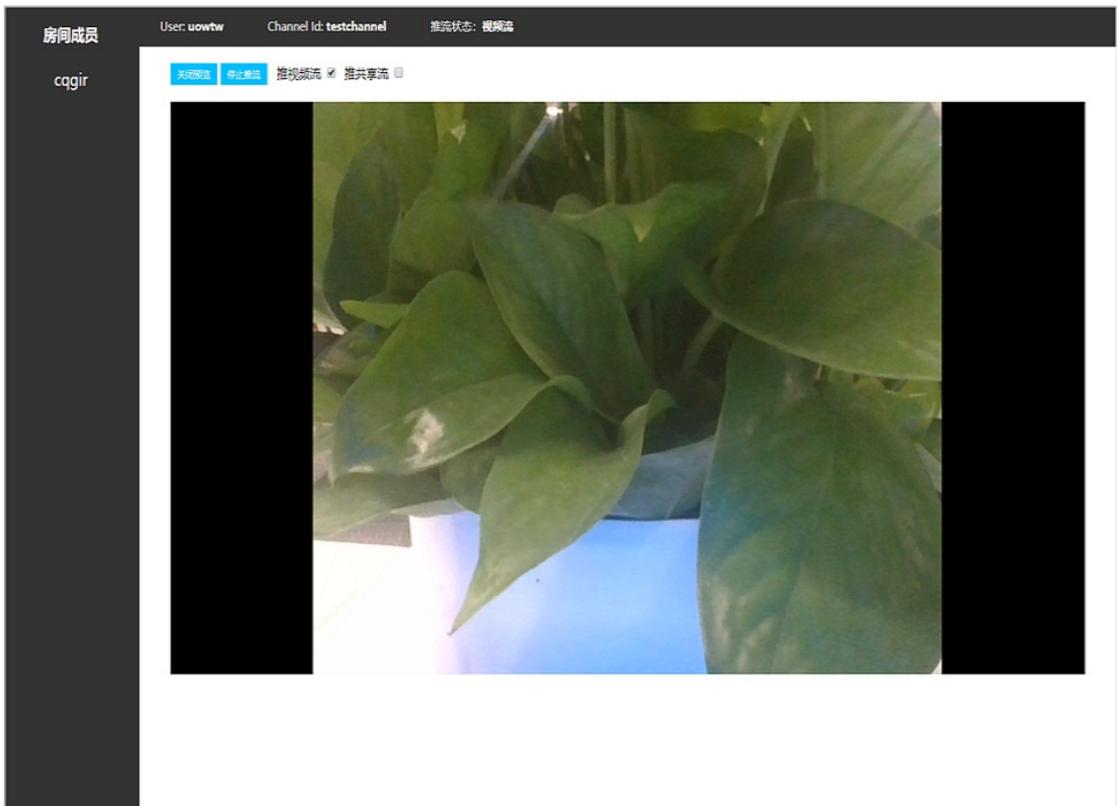
```
var appId = "yourAppId";
var appKey = "yourAppKey";
```

2. 运行Demo。

- i. 输入房间号，单击进入房间。



- ii. 进入房间后，已经默认开启本地预览并开始推送视频流（摄像头大流和音频）。您可以开启或关闭本地预览，也可以打开或关闭推流。



**?** 说明 房间显示用户名、房间号、推流状态。如果在当前房间内有其他成员，左侧会显示其用户名。

- 如果您想更改推流状态，您可以直接对推流选项进行热切换，也可以先单击**停止推流**，然后选择**推视频流**或**推共享流**。

**关闭预览** **停止推流** **推视频流**  **推共享流**

- 如果您想和其他房间成员进行音视频通信，您可以将鼠标移动到左侧该成员用户名上，当出现共享流 订阅或视频流 订阅时，然后进行选择操作。

② 说明 Demo中仅可订阅视频流和共享流，音频流单独订阅您可以自行开发实现，接口说明请参见[AliRtcEngine接口](#)。



② 说明 订阅后的用户界面显示在房间右侧，互相订阅后即可开始实时音视频通信。

# 7.Unity Demo

本章节为您介绍Unity Demo的集成操作步骤。

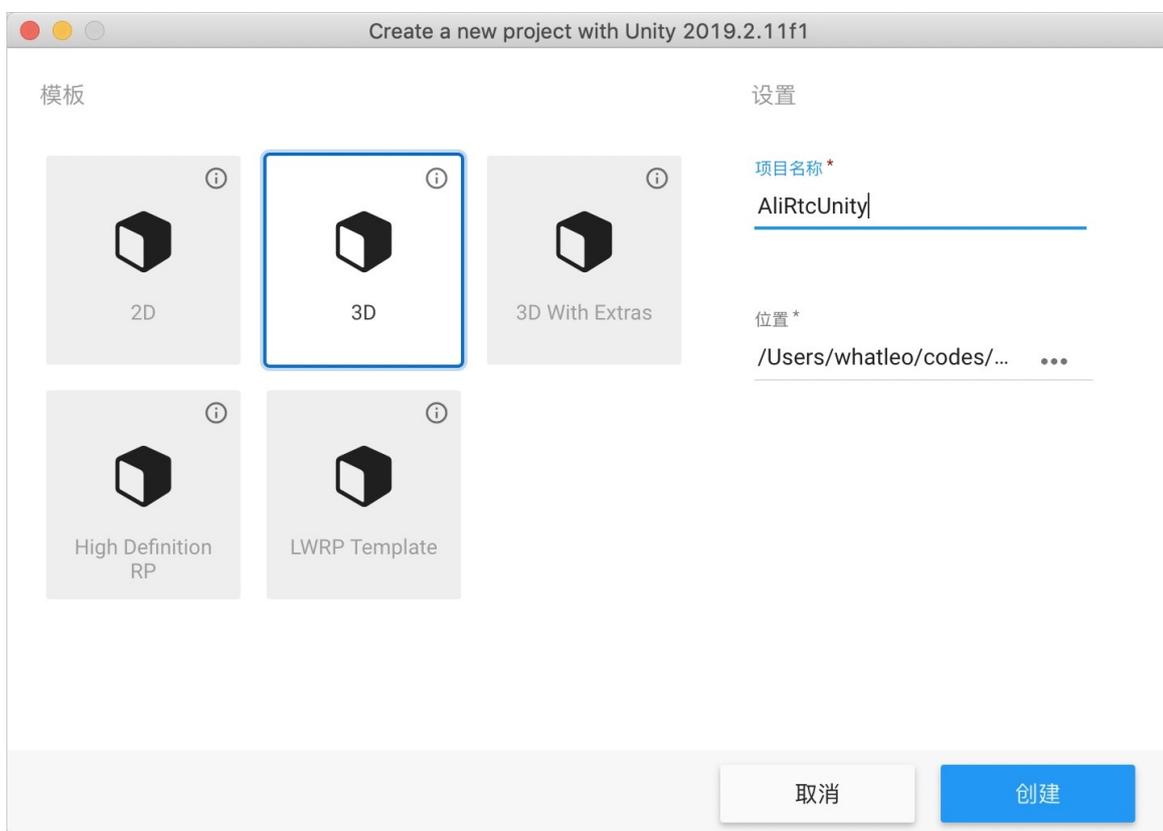
## 前提条件

您需要开通阿里云RTC服务并在控制台成功创建应用，具体操作请参见[入门概述](#)。

获取Token，详情请参见[服务端生成Token](#)。

## 创建Unity项目

1. 打开Unity，单击新建。
2. 输入项目名称、项目保存位置，并选择3D模版。
3. 单击创建。



## 集成SDK

1. 下载Unity SDK。解压后包含sample和sdk两个目录，sample目录为Unity对接工程，sdk目录为Unity版本SDK。
2. 在项目Assets目录下新建Plugins目录。
3. 复制SDK中以下目录文件到项目Plugins目录。

平台	文件或文件夹	项目路径
	/Android/AliRtcWrapper.aar	/Assets/Plugins/Android

平台	文件或文件夹	项目路径
Android	/Android/AlIRTCsdk.jar	/Assets/Plugins/Android
	/Android/alivc-core-rtc.aar	/Assets/Plugins/Android
	/Android/Sophonsdk.aar	/Assets/Plugins/Android
	/Android/utdid4all-1.5.0-proguard.jar	/Assets/Plugins/Android
	/Android/webrtclicb.aar	/Assets/Plugins/Android
iOS	/iOS/AlIRTCsdk.framework	/Assets/Plugins/iOS
	/iOS/libAlIRTCsdkCInterface.a	/Assets/Plugins/iOS
macOS	/macOS/AlIRTCsdkCWrapper.bundle	/Assets/Plugins/macOS
Windows	/x86/AlIRTCsdk.dll	/Assets/Plugins/x86
	/x86/AlIRTCsdkCWrapper.dll	/Assets/Plugins/x86
	/x64/AlIRTCsdk.dll	/Assets/Plugins/x64
	/x64/AlIRTCsdkCWrapper.dll	/Assets/Plugins/x64

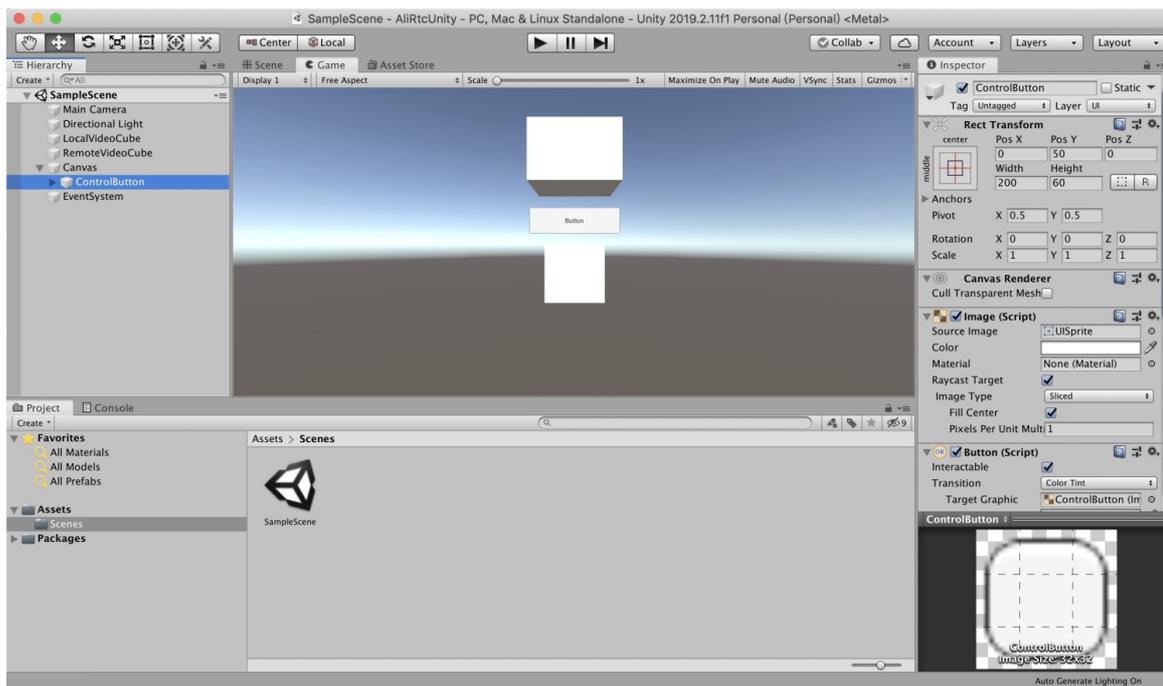
4. 在项目Assets目录下新建Scripts目录。
5. 复制SDK中以下目录文件到项目Scripts目录。

文件	项目路径
/Scripts/AlIRtcEngine.cs	/Assets/Scripts
/Scripts/AlIRtcEngineBase.cs	/Assets/Scripts
/Scripts/VideoDisplaySurface.cs	/Assets/Scripts

## 实现音视频通话功能

1. 添加UI控制。

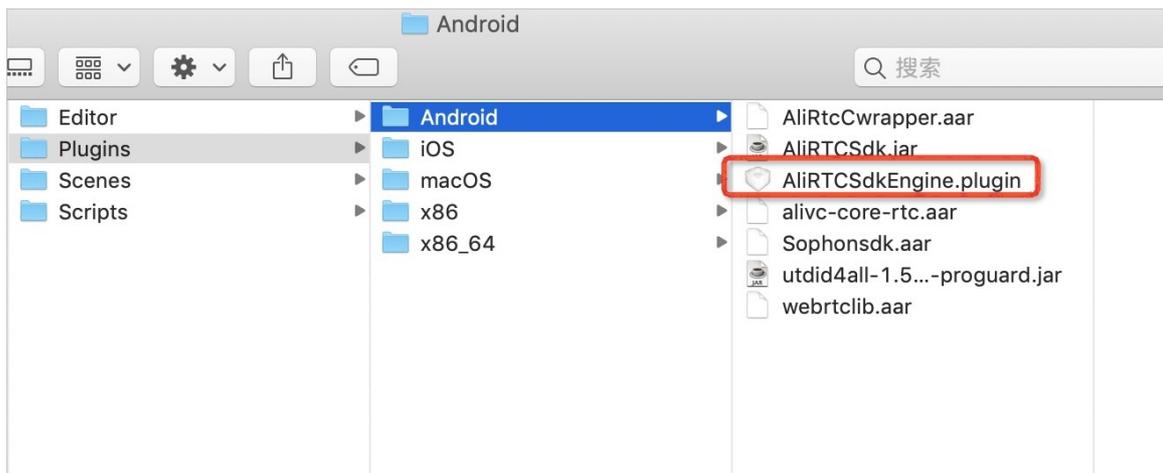
您可以创建用户界面，以Unity Sample为例，创建本地视频窗口GameObject（LocalVideoCube）和远端视频窗口GameObject（RemoteVideoCube），以及控制按钮ControlButton。



2. UI界面保存到 `/Assets/Scenes/SampleScene.unity`。
3. 创建控制音视频通信功能的 `Home.cs` 文件，并将该文件添加到主 Main Camera，使得 `Home.cs` 文件在 Main Camera 启动时就被加载。
4. 获取权限（仅 Android 操作）。在 UNITY\_2018\_3\_OR\_NEWER 及以上版本，Android 设备需要在 Unity 中设置主动向用户获取麦克风和相机权限，需要调用 `CheckPermission` 主动获取权限。

```
private ArrayList permissionList = new ArrayList();
void Start ()
{
    #if (UNITY_2018_3_OR_NEWER && UNITY_ANDROID)
        permissionList.Add(Permission.Microphone);
        permissionList.Add(Permission.Camera);
    #endif
}
void Update ()
{
    #if (UNITY_2018_3_OR_NEWER && UNITY_ANDROID)
        CheckPermission();
    #endif
}
private void CheckPermission()
{
    #if (UNITY_2018_3_OR_NEWER && UNITY_ANDROID)
        foreach (string permission in permissionList)
        {
            if (Permission.HasUserAuthorizedPermission(permission))
            {
            }
            else
            {
                Permission.RequestUserPermission(permission);
            }
        }
    #endif
}
```

此外，集成Android平台时，需要AndroidManifest.xml文件和project.properties文件来管理项目权限和项目属性，可以直接拷贝AliRTCSDKEngine.plugin目录到/Assets/Plugins/Android。



5. 初始化IAliRtcEngine。

执行创建IAliRtcEngine，根据需要注册侦听通知回调，设置自动推流订阅模式。

```
string extra = "";
IAliRtcEngine mRtcEngine = IAliRtcEngine.GetEngine (extra);
mRtcEngine.OnJoinChannelNotify = onJoinChannelNotify;
mRtcEngine.OnPublishNotify = onPublishNotify;
mRtcEngine.OnSubscribeNotify = onSubscribeNotify;
mRtcEngine.OnRemoteUserOnLineNotify = onRemoteUserOnLineNotify;
mRtcEngine.OnRemoteUserOffLineNotify = onRemoteUserOffLineNotify;
mRtcEngine.OnRemoteTrackAvailableNotify = onRemoteTrackAvailableNotify;
mRtcEngine.OnSubscribeChangedNotify = onSubscribeChangedNotify;
mRtcEngine.OnLeaveChannelResultNotify = onLeaveChannelResultNotify;
mRtcEngine.SetAutoPublish(true, true);
```

6. 打开本地预览。

进入频道前需要启动本地视频采集预览显示，获取UI中VideoCube对象，添加VideoDisplaySurface到VideoCube上。

**说明** VideoDisplaySurface是SDK封装用于视频渲染的C#类，任何Unity中GameObject物体上需要显示视频内容，必须要绑定VideoDisplaySurface。

VideoDisplaySurface主要有以下三个功能：

- void SetUserId(string uid): 设置渲染视频的用户ID，本地uid填空字符串 ""。
- void SetVideoTrack(AliRTCVideoTrack track): 设置视频流类型，有camera（相机流）和screen（共享流）两种，用于区分同一用户ID推送的两种不同的是视频流。
- void SetEnable(bool enable): 控制视频是否显示。

添加surface后，调用IAliRtcEngine接口设置本地预览和开启预览。

```
GameObject go = GameObject.Find("LocalVideoCube");
VideoDisplaySurface surface = go.GetComponent<VideoDisplaySurface>();
surface.SetUserId("");
surface.SetVideoTrack(AliRTCVideoTrack.VIDEO_TRACK_CAMERA);
surface.SetEnable(true);
mRtcEngine.SetLocalViewConfig(true);
mRtcEngine.ConfigExternalVideoRendering(true);
mRtcEngine.StartPreview();
```

7. 获取频道鉴权信息。该步骤主要通过和AppServer进行通信，获取进入的频道名称以及对应的鉴权信息，成功获取到的信息需要保存在AliRTCAuthInfo结构中。
8. 加入频道。成功获取到入会鉴权信息后，通过调用接口JoinChannel加入频道，需要传递参数有AliRTCAuthInfo和用户名。调用加入频道接口后，SDK会通过OnJoinChannelNotify通知回调是否入会成功，参数errorCode为0表示入会成功，其他表示失败。

```
mRtcEngine.JoinChannel(authInfo, "userName");
private void onJoinChannelNotify (int errorCode)
{
    if (errorCode == 0)
    {
        Debug.Log("加入频道成功");
    }
    else
    {
        Debug.Log("加入频道失败");
    }
}
```

9. 远端用户视频显示。

订阅远端用户的音视频流相关的回调是onSubscribeChangedNotify，控制远端用户视频的显示在该回调中实现。

onSubscribeChangedNotify有三个参数：

- o userId表示远端用户ID。
- o audioTrack有mic（麦克风）和none两种类型，none表示当前没有远端音频流。
- o videoTrack有camera、screen、both和none四种类型，分别表示该用户的视频流状态，both表示同时订阅远端用户的camera（相机流）和screen（共享流），none表示当前没有远端视频流。

在回调中，检查参数videoTrack是否为camera、screen、both三者中的一种，如果是需要使用GameObject对象进行显示，并添加VideoDisplaySurface，如果videoTrack为none，需要控制GameObject对象取消显示。

```
private void onSubscribeChangedNotify (string userId, int audioTrack, int videoTrack)
{
    if (videoTrack == (int)AliRTCVideoTrack.VIDEO_TRACK_CAMERA)
    {
        GameObject go = GameObject.Find("RemoteVideoCube");
        if (!ReferenceEquals(go, null))
        {
            VideoDisplaySurface surface = go.GetComponent<VideoDisplaySurface>();
            surface.SetUserId(userId);
            surface.SetVideoTrack(AliRTCVideoTrack.VIDEO_TRACK_CAMERA);
            surface.SetEnable(true);
        }
    }
    else if (videoTrack == (int)AliRTCVideoTrack.VIDEO_TRACK_NONE)
    {
        GameObject go = GameObject.Find("RemoteVideoCube");
        if (!ReferenceEquals(go, null))
        {
            VideoDisplaySurface surface = go.GetComponent<VideoDisplaySurface>();
            surface.SetEnable(false);
        }
    }
}
```

10. 离开频道。结束音视频通话时调用接口LeaveChannel离开频道，离开频道前需要关闭本地预览以及移除远端视频显示。

```
mRtcEngine.StopPreview();
GameObject go = GameObject.Find("LocalVideoCube");
VideoDisplaySurface surface = go.GetComponent<VideoDisplaySurface>();
surface.SetEnable(false);
go = GameObject.Find("RemoteVideoCube");
surface = go.GetComponent<VideoDisplaySurface>();
surface.SetEnable(false);
mRtcEngine.LeaveChannel();
```

11. 销毁IAliRtcEngine。

调用IAliRtcEngine的Destroy方法退出App或者释放内存。

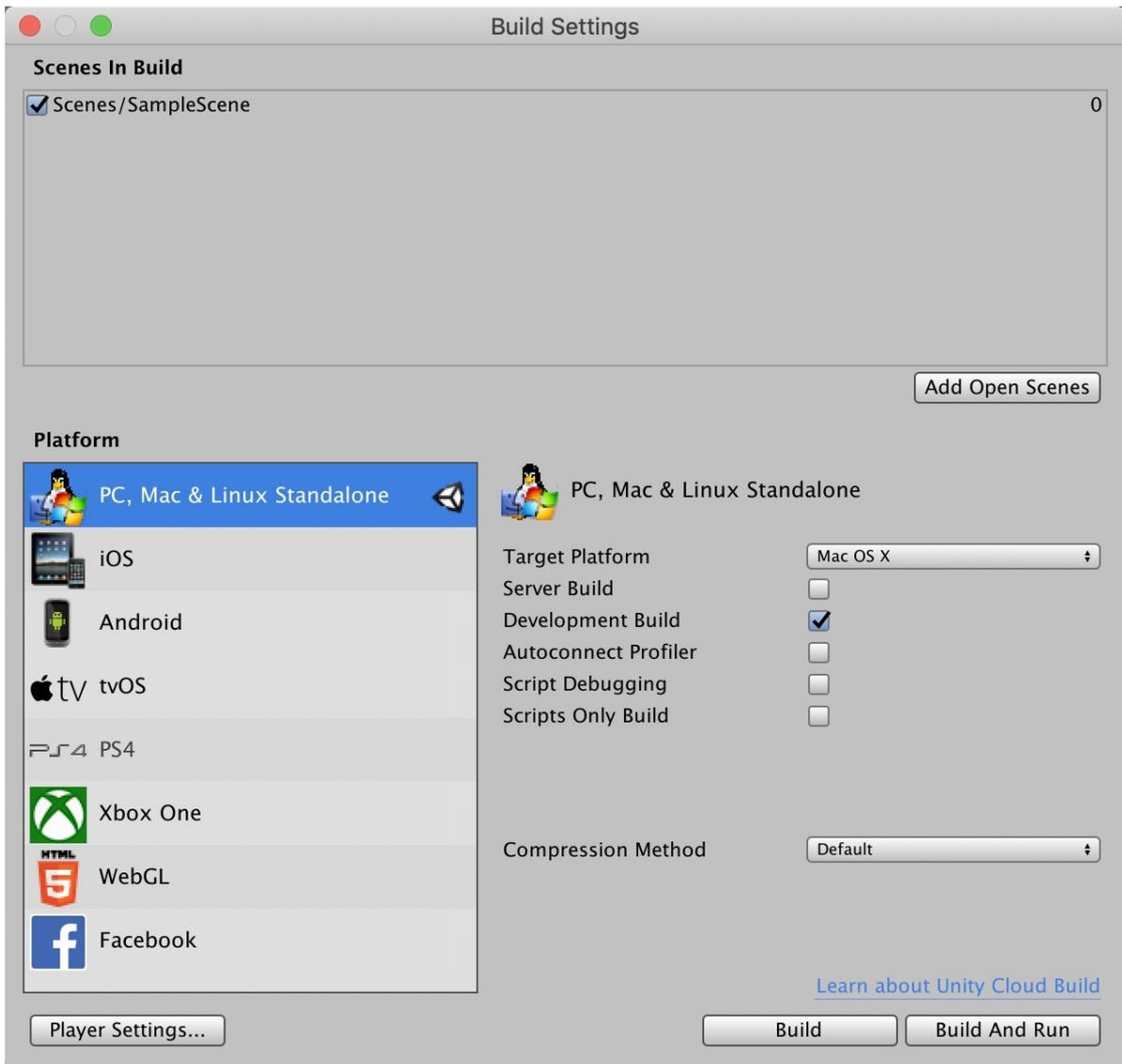
## Unity Editor中调试

Unity Editor中调试只需要确保放入macOS或Windows所需的文件或文件夹到/Assets/Plugins目录即可。

## 工程导出设置

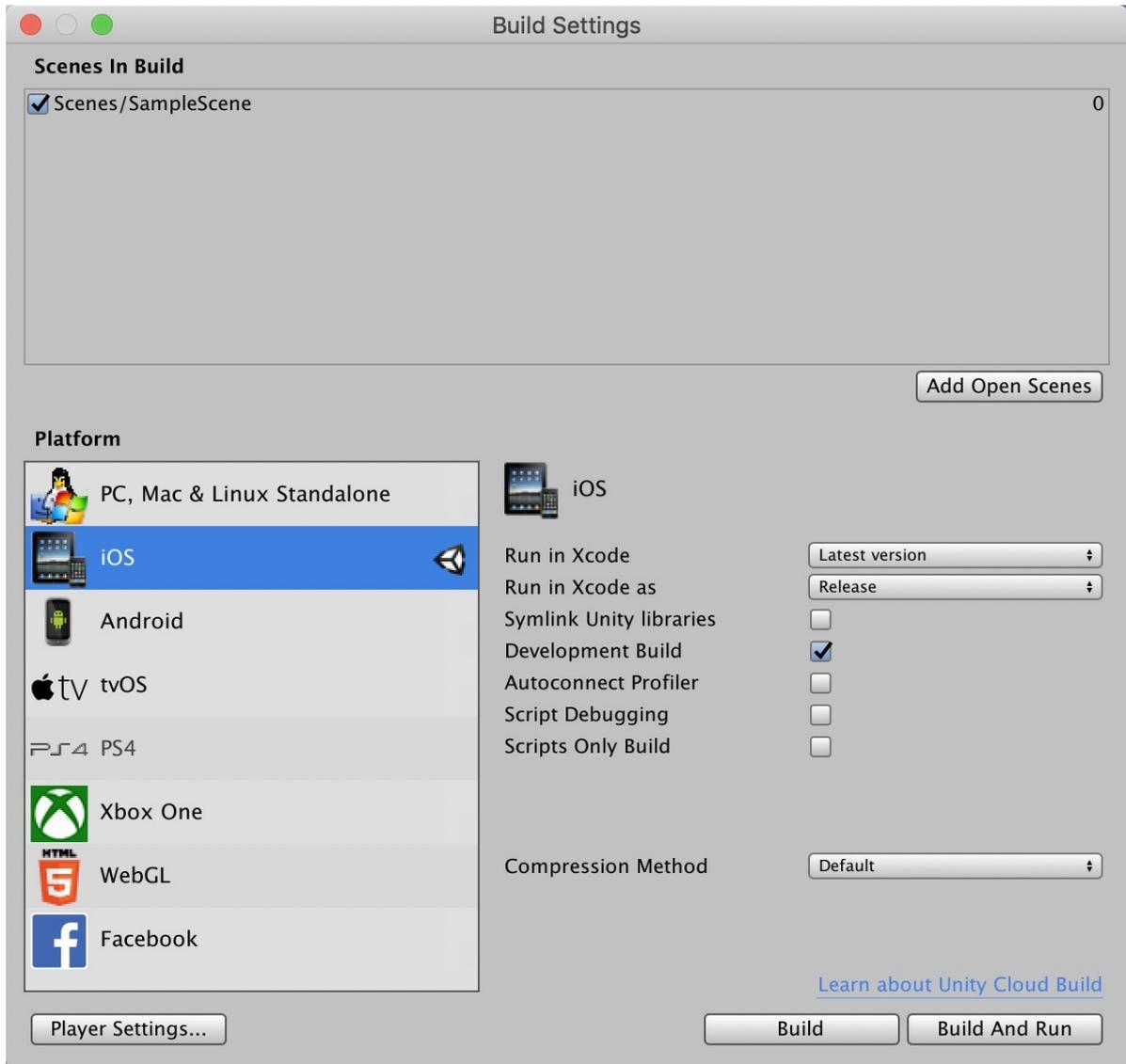
- Mac平台：

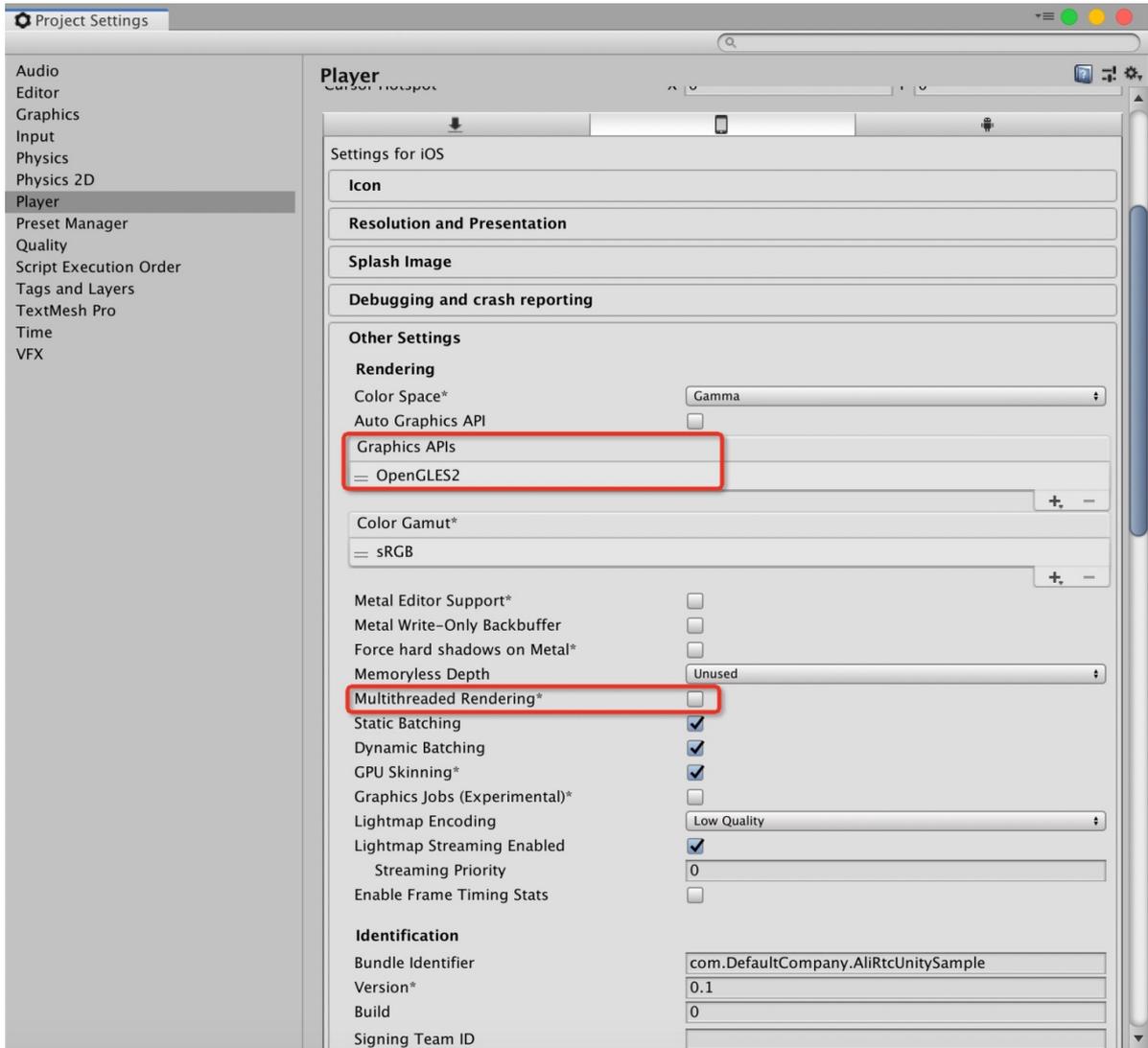
Unity编译导出Mac平台应用， macOS10.15以下版本无需做额外设置， 只需选中Target Platform为Mac OS， 勾选Development Build即可； macOS10.15及以上版本需要选择Player Settings， 在Player的Other Settings选项中， 将Camera Usage Description文本框中添加open camera字符串， Microphone Usage Description文本框中添加open microphone字符串。



- iOS平台：

设置Graphics APIs为OpenGL ES2， 去掉多线程渲染Multithreaded Rendering。

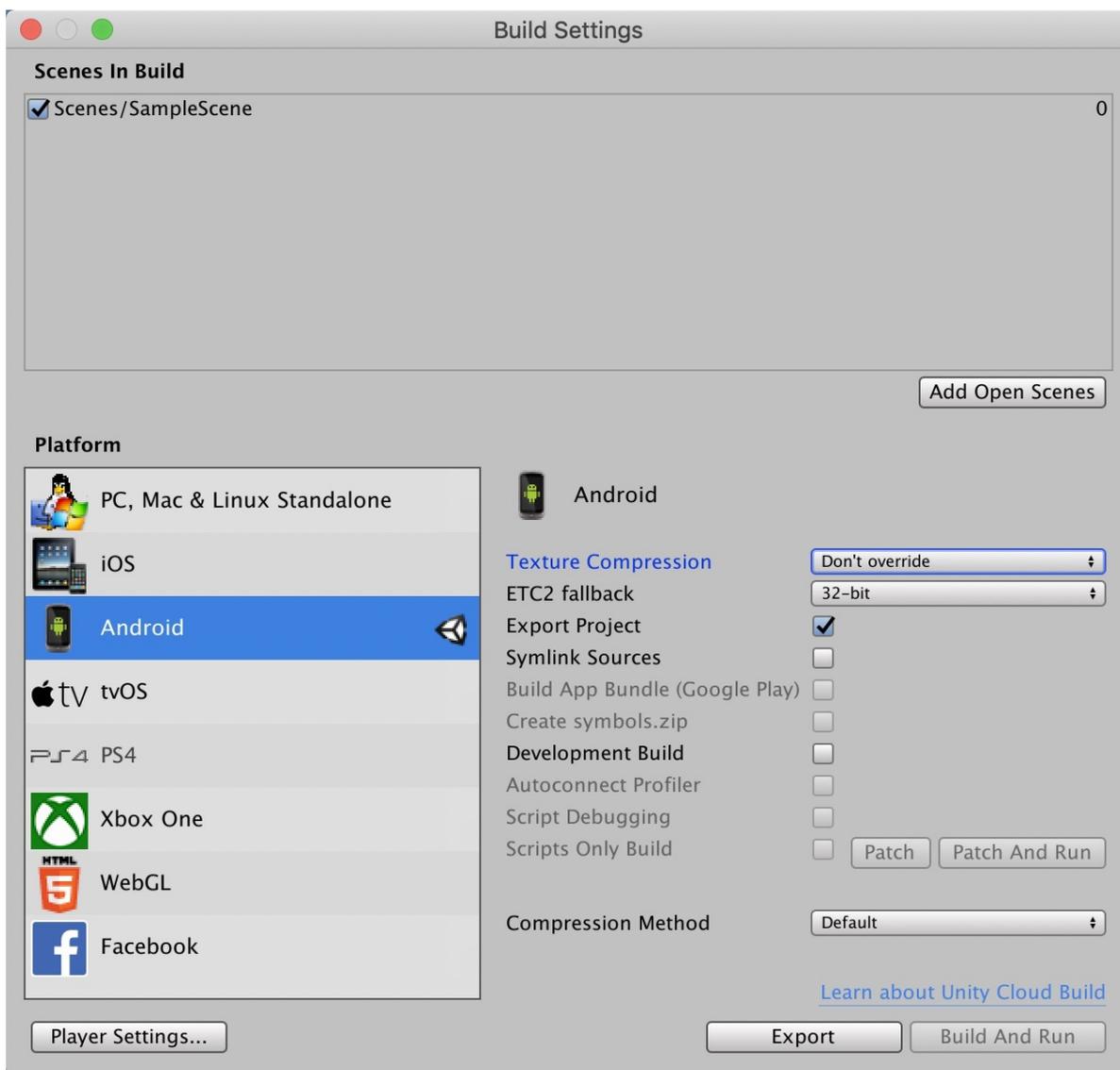


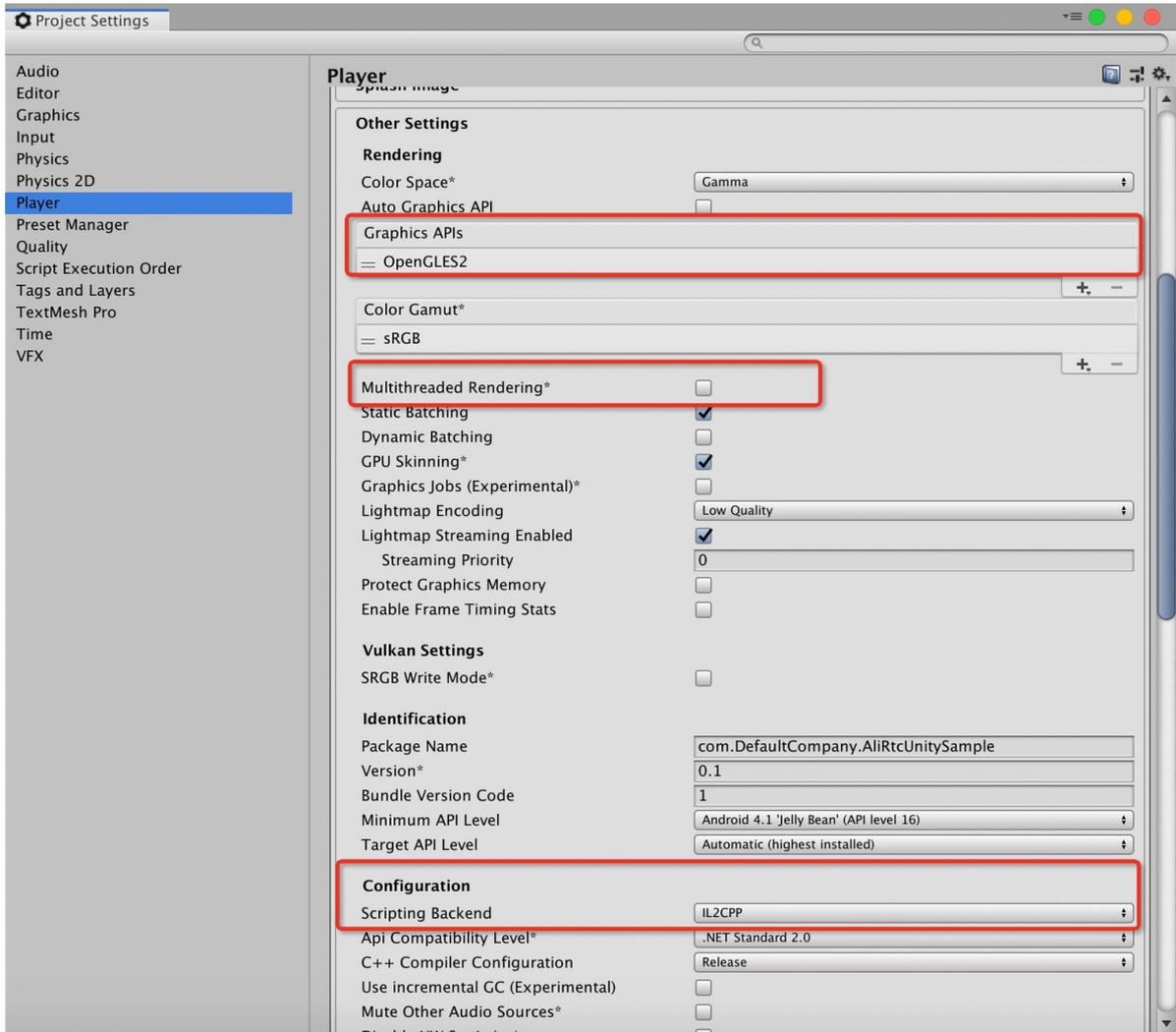


iOS导出为xcode工程，在xcode工程中需要先关闭bitcode，在`info.plist`文件中添加麦克风和相机访问，并且在Embed Framework中添加`AliRtcSdk.framework`，开启background模式，以上设置也可以在Unity打包后处理脚本中添加导出的工程设置。

- Android平台：

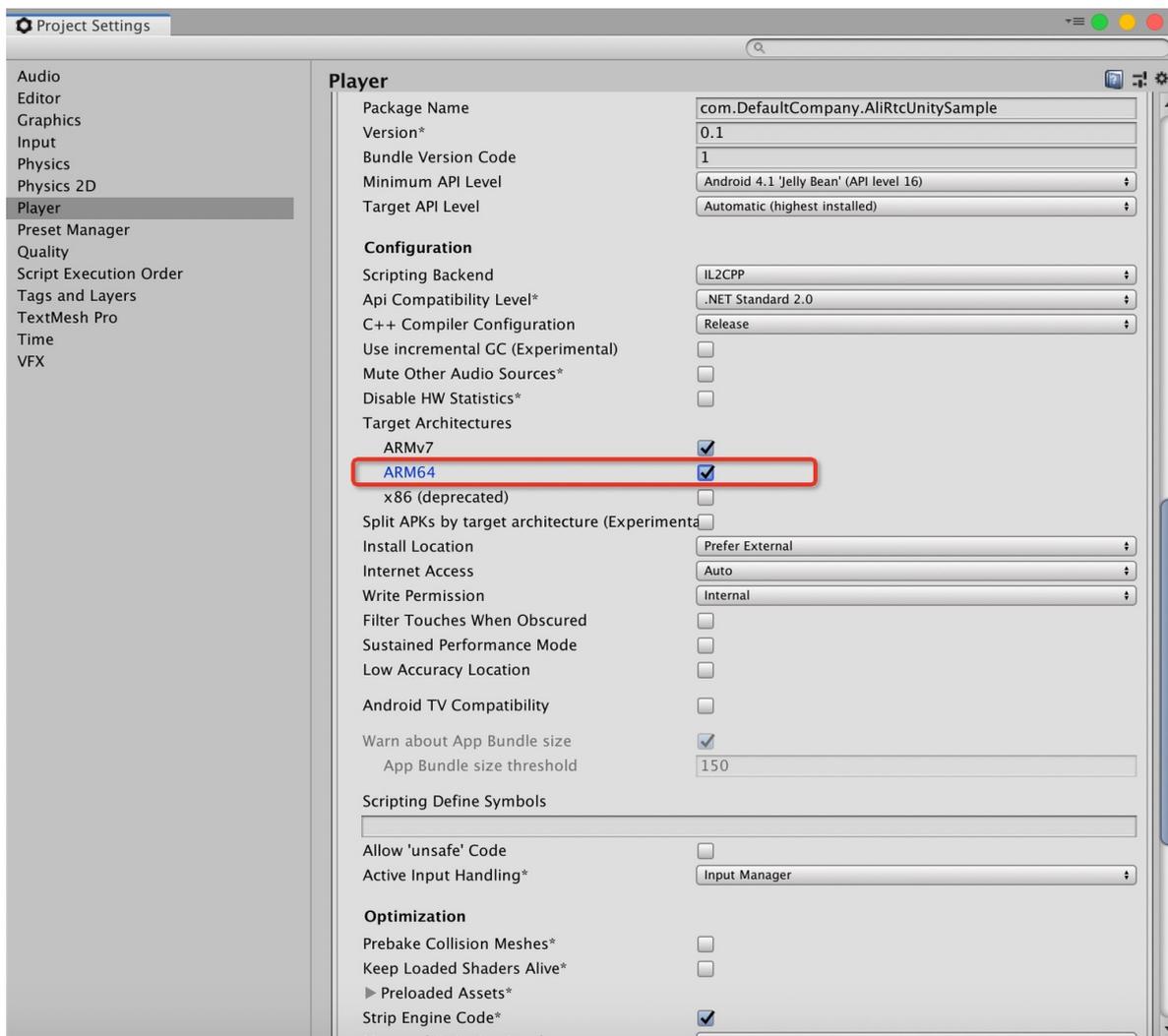
Android应用导出前，切换到Android Platform，选择Player Settings，在Player的Other Settings选项中，设置Graphics APIs为OpenGL ES 2，去掉多线程渲染Multithreaded Rendering，Scripting Backend选择IL2CPP。





说明 默认Android打包ARMv7版本，如果打包ARM64版本，需要选中Target Architectures中ARM64选项。

选中该Export Project后导出目录需要有子级文件夹，例如可导出在以下目录：*Aliyun/UnityRTC*。



- Windows平台：

Windows应用导出前，切换到PC Platform，选择Player Settings，在Player的Scripting Backend选项选中Mono。

**Scenes In Build**

Add Open Scenes

**Platform**

- PC, Mac & Linux Standalone
- Universal Windows Platform
- tvOS
- PS4
- iOS
- Xbox One
- Android
- WebGL

**PC, Mac & Linux Standalone**

Target Platform: Windows

Architecture: x86

Server Build:

Copy PDB files:

Create Visual Studio Solution:

Development Build:

Autoconnect Profiler:

Script Debugging:

Scripts Only Build:

Compression Method: Default

[Learn about Unity Cloud Build](#)

Player Settings... Build Build And Run