ALIBABA CLOUD

阿里云

音视频通信

常见问题

文档版本：20220701

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**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.概览 05
2.集成问题 06
2.1. 集成Android SDK时的常见问题 06
2.2. 集成iOS SDK时的常见问题 07
2.3. 集成Mac SDK时的常见问题 12
2.4. 集成Windows SDK时的常见问题 13
2.5. 集成Web SDK时的常见问题 15
2.6. Android SDK动态加载so文件方案 16
3.质量问题 19
3.1. 客户端入会失败常见原因 19
3.2. 客户端与服务端连通异常 19
3.3. H5纯订阅模式媒体文件播放失败 19
4.SDK使用问题 24
4.1. 如何实现多模块共享AVAudioSession 24
4.2. SDK使用过程中如何减少耗时 24
4.3. Web端没有麦克风设备如何推视频流 25
4.4. 通讯模式升级至互动模式说明 25
4.5. 视频双流规格配置说明 26
4.6. 小程序SDK常见问题 32
5.其他问题 35
5.1. H5端如何实现镜像及显示横屏录制的视频 35
5.2. 旁路转推纯音频的配置说明 35
5.3. 如何进行角色鉴权 35

1.概览

集成问题

- 集成Android SDK时的常见问题
- 集成iOS SDK时的常见问题
- 集成Mac SDK时的常见问题
- 集成Windows SDK时的常见问题
- 集成Web SDK时的常见问题
- Android SDK动态加载so文件方案

质量问题

- 客户端入会失败常见原因
- 客户端与服务端连通异常
- H5纯订阅模式媒体文件播放失败

SDK使用问题

- 如何实现多模块共享AVAudioSession
- SDK使用过程中如何减少耗时
- 手动订阅和手动推流说明
  - Android端手动订阅和手动推流
  - Windows端手动订阅和手动推流
- 1对1语音聊天
  - 服务端运行常见问题
  - iOS端运行常见问题
  - Android端运行常见问题
- Web端没有麦克风设备如何推视频流
- 通讯模式升级至互动模式说明

其他问题

- H5端如何实现镜像及显示横屏录制的视频
- 旁路转推纯音频的配置说明
- 如何进行角色鉴权

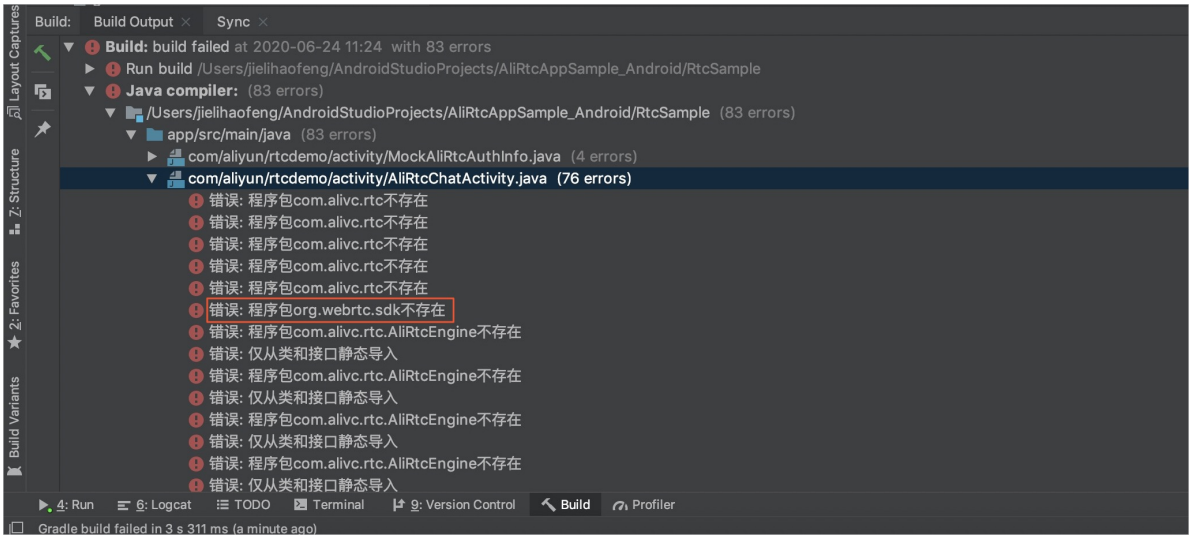
2.集成问题

2.1. 集成Android SDK时的常见问题

通过阅读本文，您可以了解集成Android SDK时常见的问题及解决方法。

gradle中未正确引用Android SDK依赖文件导致编译报错

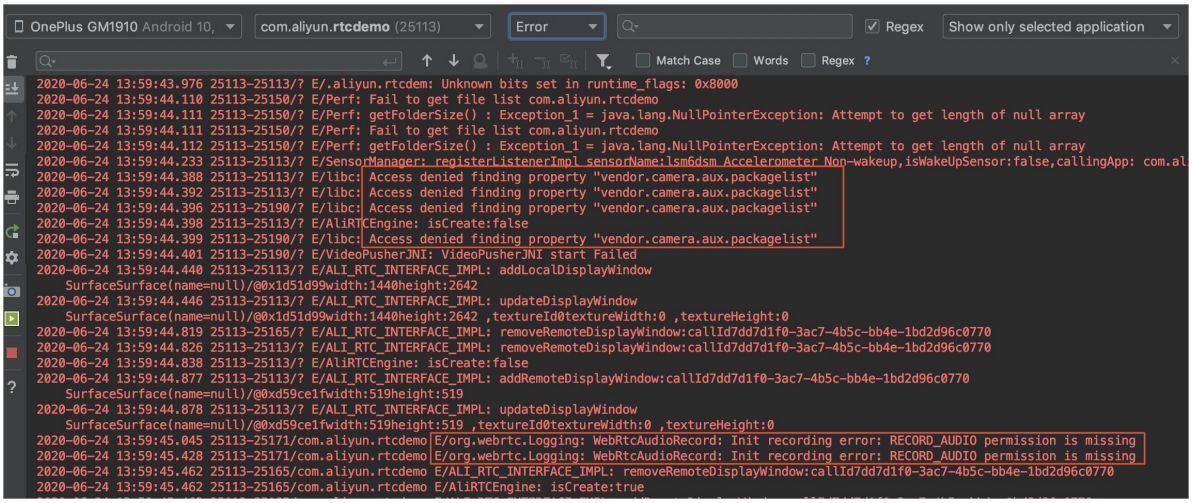
- 问题现象：编译代码时可能会出现以下错误：



- 可能原因：gradle中未正确引用Android SDK依赖文件。
- 解决方案：请按照正确步骤导入aar包和jar包，并在gradle中配置引用。具体操作，请参见集成Android SDK。

隐私权限未申请导致程序运行时报错

- 问题现象：程序运行时可能会出现以下错误：



- 可能原因：隐私权限未申请。
- 解决方案：

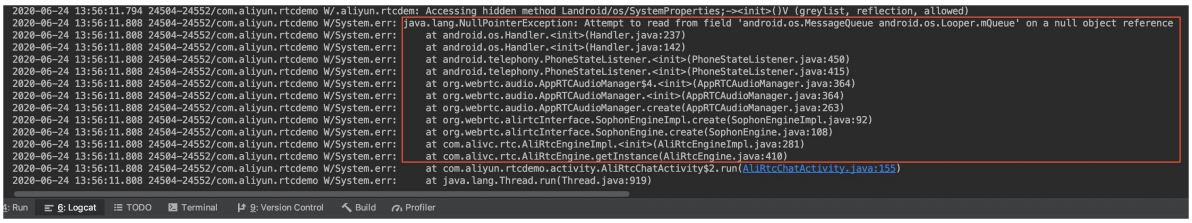
- 在*AndroidManifest.xml*文件中添加摄像头、麦克风、网络、访问存储等权限。

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS"/>
```

- 在代码里动态申请权限。

未在主线程初始化SDK导致程序运行时报错

- 问题现象：程序运行时可能会出现以下错误：



- 可能原因：未在主线程初始化实例。
- 解决方案：初始化 `AliRtcEngine` 实例，并注册回调。相关回调有 `AliRtcEngineEventListener` 和 `AliRtcEngineNotify` ，并且只能在主线程调用。

```
engine = AliRtcEngine.getInstance(getApplicationContext());
engine.setRtcEngineEventListener(mEventListener);
engine.setRtcEngineNotify(mEngineNotify);
```

更多回调接口详情，请参见回调及监听。

弱网情况下人声有卡顿

- 问题现象：在线KTV场景下，弱网情况下人声有卡顿。
- 可能原因：为了保证合唱实时性，客户端采用了低延时策略，弱网下丢包率会相应增加。

开启耳返模式下声音外放有回声

- 问题现象：在线KTV场景下，开启耳返模式下声音外放有回声。
- 解决方案：您需要带上耳机然后进行合唱，不能通过外放。

2.2. 集成iOS SDK时的常见问题

通过阅读本文，您可以了解集成iOS SDK时常见的问题及解决方法。

编译代码时报x86或i386错误

- 问题现象：编译代码时可能会出现以下错误：

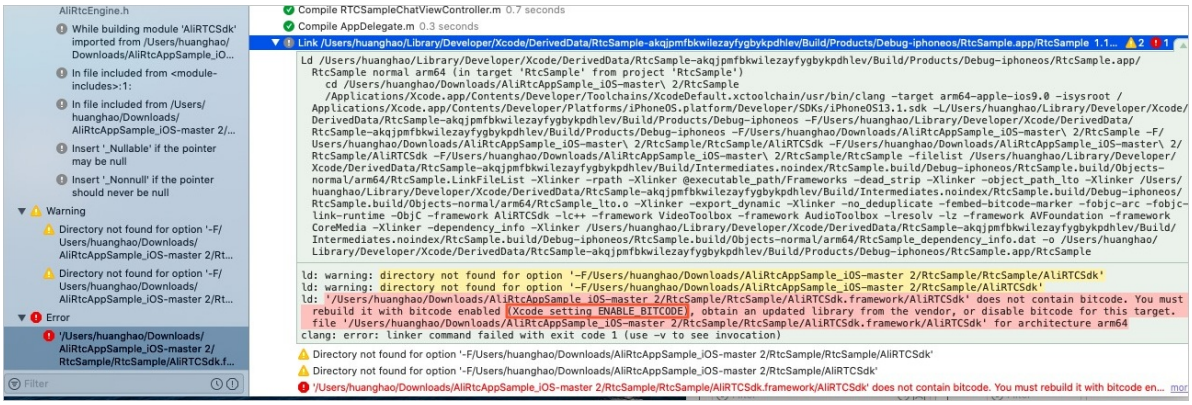
```
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/clang -target
x86_64-apple-ios9.0-simulator -isysroot /Applications/Xcode.app/Contents/Developer/Platforms/
iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator13.1.sdk -L/Users/huanghao/Library/Developer/
Xcode/DerivedData/RtcSample-akqjpmfbkwilezayfygbykpdhlev/Build/Products/Debug-iphonesimulator -F/Users/
huanghao/Library/Developer/Xcode/DerivedData/RtcSample-akqjpmfbkwilezayfygbykpdhlev/Build/Products/
Debug-iphonesimulator -F/Users/huanghao/Downloads/AliRtcAppSample_iOS-master\ 2/RtcSample -F/Users/
huanghao/Downloads/AliRtcAppSample_iOS-master\ 2/RtcSample/RtcSample/AliRTCSdk -F/Users/huanghao/
Downloads/AliRtcAppSample_iOS-master\ 2/RtcSample/AliRTCSdk -F/Users/huanghao/Downloads/
AliRtcAppSample_iOS-master\ 2/RtcSample/RtcSample -filelist /Users/huanghao/Library/Developer/Xcode/
DerivedData/RtcSample-akqjpmfbkwilezayfygbykpdhlev/Build/Intermediates.noindex/RtcSample.build/Debug-
iphonesimulator/RtcSample.build/Objects-normal/x86_64/RtcSample.LinkFileList -Xlinker -rpath -Xlinker
@executable_path/Frameworks -dead_strip -Xlinker -object_path_lto -Xlinker /Users/huanghao/Library/
Developer/Xcode/DerivedData/RtcSample-akqjpmfbkwilezayfygbykpdhlev/Build/Intermediates.noindex/
RtcSample.build/Debug-iphonesimulator/RtcSample.build/Objects-normal/x86_64/RtcSample_lto.o -Xlinker
-export_dynamic -Xlinker -no_deduplicate -Xlinker -objc_abi_version -Xlinker 2 -fobjc-arc -fobjc-link-
runtime -ObjC -Xlinker -sectcreate -Xlinker __TEXT -Xlinker __entitlements -Xlinker /Users/huanghao/
Library/Developer/Xcode/DerivedData/RtcSample-akqjpmfbkwilezayfygbykpdhlev/Build/Intermediates.noindex/
RtcSample.build/Debug-iphonesimulator/RtcSample.build/RtcSample.app-Simulated.xcent -framework AliRTCSdk
-lc++ -framework VideoToolbox -framework AudioToolbox -lresolv -lz -framework AVFoundation -framework
CoreMedia -Xlinker -dependency_info -Xlinker /Users/huanghao/Library/Developer/Xcode/DerivedData/
RtcSample-akqjpmfbkwilezayfygbykpdhlev/Build/Intermediates.noindex/RtcSample.build/Debug-
iphonesimulator/RtcSample.build/Objects-normal/x86_64/RtcSample_dependency_info.dat -o /Users/huanghao/
Library/Developer/Xcode/DerivedData/RtcSample-akqjpmfbkwilezayfygbykpdhlev/Build/Products/Debug-
iphonesimulator/RtcSample.app/RtcSample

ld: warning: directory not found for option '-F/Users/huanghao/Downloads/AliRtcAppSample_iOS-master 2/
  RtcSample/RtcSample/AliRTCSdk'
ld: warning: directory not found for option '-F/Users/huanghao/Downloads/AliRtcAppSample_iOS-master 2/
  RtcSample/AliRTCSdk'
ld: warning: ignoring file /Users/huanghao/Downloads/AliRtcAppSample_iOS-master 2/RtcSample/RtcSample/
  AliRTCSdk.framework/AliRTCSdk, missing required architecture x86_64 in file /Users/huanghao/Downloads/
  AliRtcAppSample_iOS-master 2/RtcSample/RtcSample/AliRTCSdk.framework/AliRTCSdk (2 slices)
Undefined symbols for architecture x86_64:
  "_OBJC_CLASS_$_AliVideoCanvas", referenced from:
      objc-class-ref in RTCSampleChatViewController.o
  "_OBJC_CLASS_$_AliRtcAuthInfo", referenced from:
      objc-class-ref in RTCSampleChatViewController.o
  "_OBJC_CLASS_$_AliRtcEngine", referenced from:
      objc-class-ref in RTCSampleChatViewController.o
  "_OBJC_CLASS_$_AliRenderView", referenced from:
      objc-class-ref in RTCSampleRemoteUserManager.o
      objc-class-ref in RTCSampleChatViewController.o
ld: symbol(s) not found for architecture x86_64
clang: error: linker command failed with exit code 1 (use -v to see invocation)
Directory not found for option '-F/Users/huanghao/Downloads/AliRtcAppSample_iOS-master 2/RtcSample/RtcSample/AliRTCSdk'
```

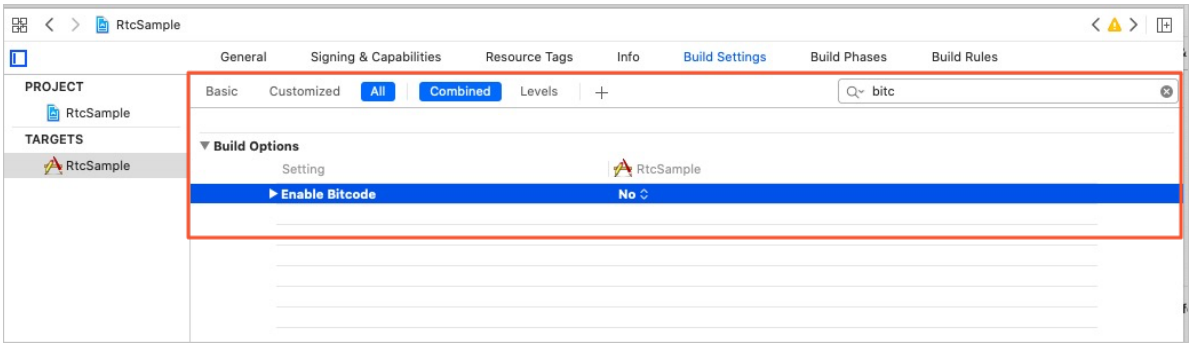
- 可能原因：使用模拟器调试和运行。
- 解决方案：请使用真实设备调试和运行。

编译代码时报bitcode错误

- 问题现象：编译代码时可能会出现以下错误：

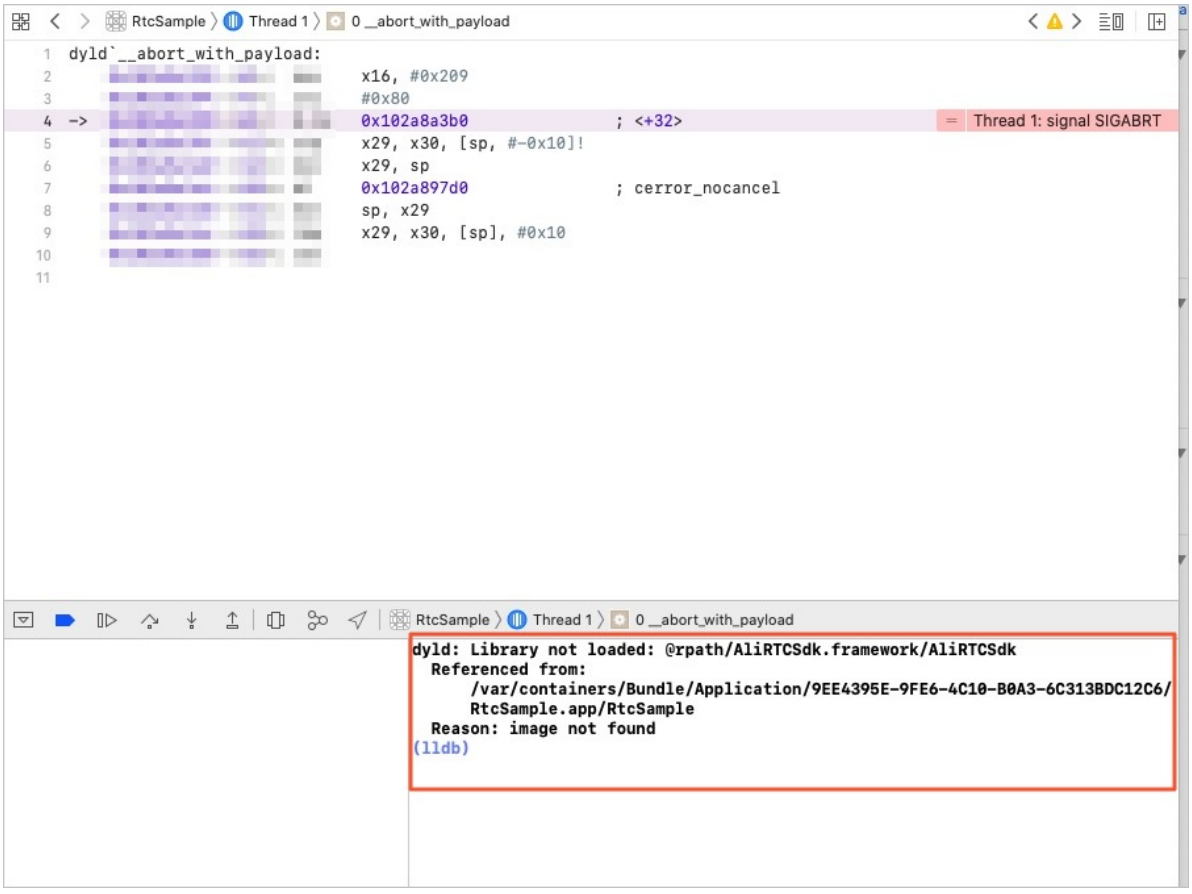


- 可能原因：SDK暂不支持bitcode配置。
- 解决方案：关闭bitcode编译选项。

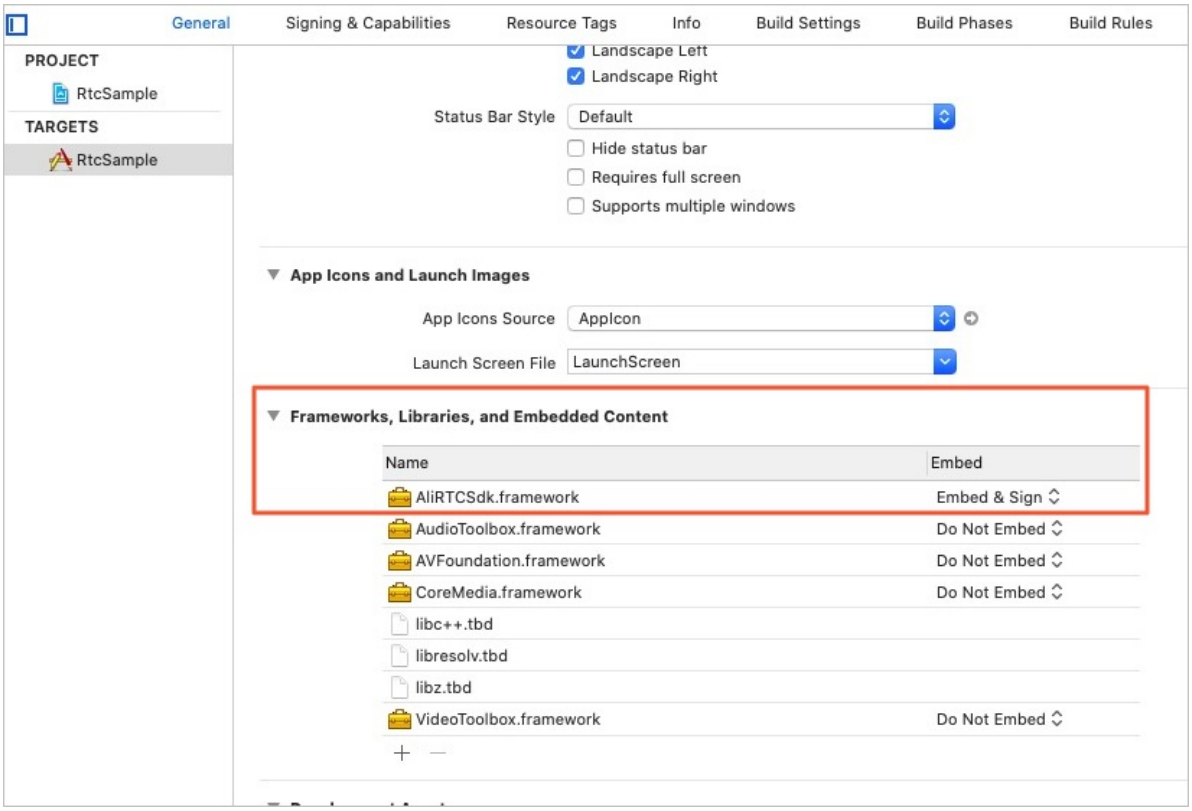


编译代码时报image not found

- 问题现象：编译代码时可能会出现以下错误：

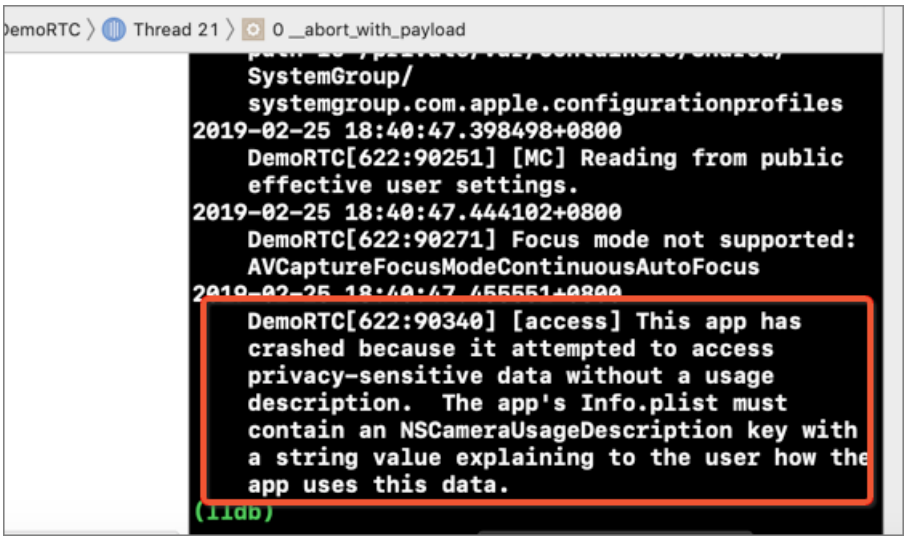


- 可能原因：SDK 1.6及之前版本是静态加载，从1.7版本开始切换为动态加载。
- 解决方案：将*AliRTCSdk.framework*加载到Embedded Binaries中。

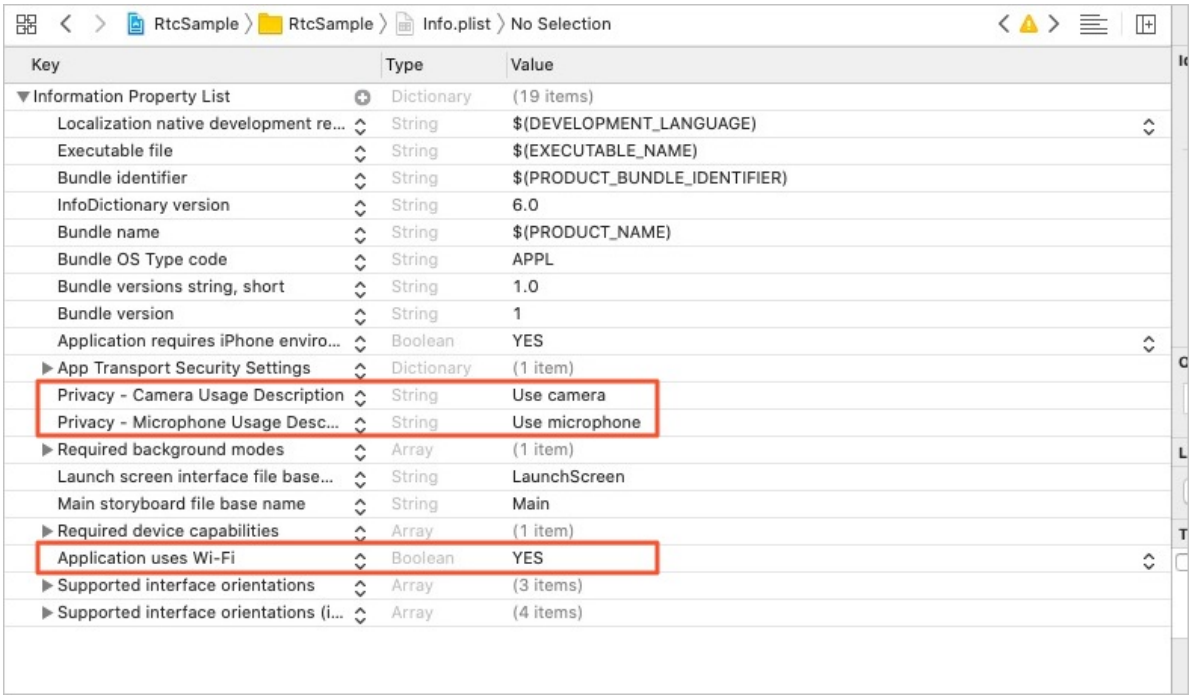


隐私权限未申请导致程序运行时报

- 问题现象：程序运行时可能会出现以下错误：

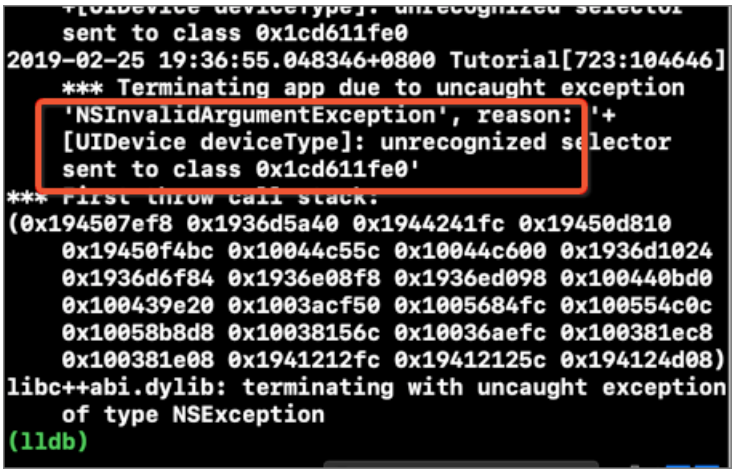


- 可能原因：隐私权限未申请。
- 解决方案：编辑*info.plist*，申请摄像头、麦克风等权限。具体操作，请参见集成iOS SDK。

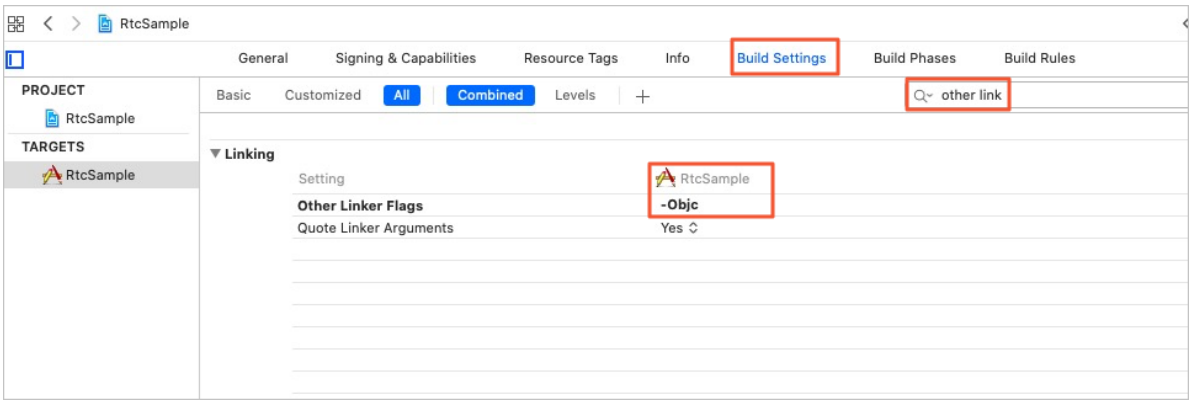


UIDevice方法调用错误

- 问题现象：程序运行时可能会出现以下错误：



- 解决方案：添加-ObjC到链接选项里。



弱网情况下人声有卡顿

- 问题现象：在线KTV场景下，弱网情况下人声有卡顿。
- 可能原因：为了保证合唱实时性，客户端采用了低延时策略，弱网下丢包率会相应增加。

开启耳返模式下声音外放有回声

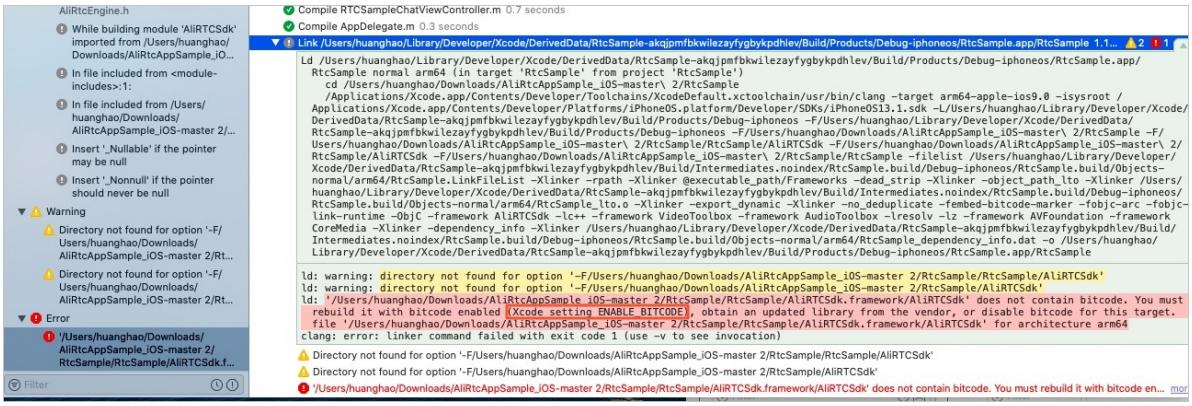
- 问题现象：在线KTV场景下，开启耳返模式下声音外放有回声。
- 解决方案：您需要带上耳机然后进行合唱，不能通过外放。

2.3. 集成Mac SDK时的常见问题

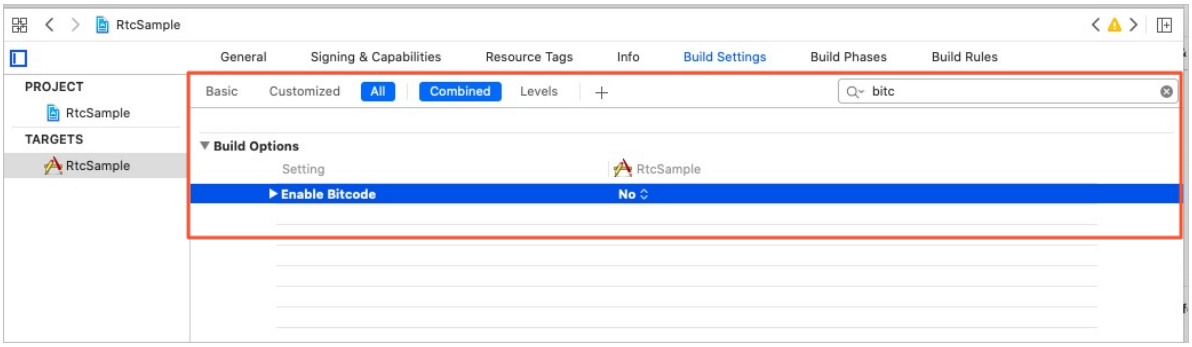
通过阅读本文，您可以了解集成Mac SDK时常见的问题及解决方法。

编译代码时报bitcode错误

- 问题现象：编译代码时可能会出现以下错误：



- 可能原因：SDK暂不支持bitcode配置。
- 解决方案：关闭bitcode编译选项。



编译代码时报image not found

- 问题现象：编译代码时可能会出现以下错误：

- 可能原因：SDK 1.6及之前版本是静态加载，从1.7版本开始切换为动态加载。
- 解决方案：将*UTDID.framework*加载到Embedded Binaries中。具体操作，请参见集成Mac SDK。

参数不符合规范

- 问题现象：程序运行时可能会出现以下错误：

```
objc[1044]: Class RTCCVPixelBuffer is implemented in both
    /System/Library/PrivateFrameworks/WebCore.framework/Frameworks/libwebrtc.dylib (0x1c141d018) and
    /private/var/containers/Bundle/Application/2B54950D-B385-4D1A-95BC-2045B77F4223/RtcSample
    .app/Frameworks/AliRTCSdk.framework/AliRTCSdk (0x103afcc68). One of the two will be used. Which
    one is undefined.
2020-06-22 14:04:36.262214+0800 RtcSample[1044:129397] Reachability Flag Status: -R -------
    networkStatusForFlags
LossBasedEstimator::ctor2020-06-22 14:04:36.273063+0800 RtcSample[1044:129397] [MC] System group
    container for systemgroup.com.apple.configurationprofiles path is
    /private/var/containers/Shared/SystemGroup/systemgroup.com.apple.configurationprofiles
2020-06-22 14:04:36.273300+0800 RtcSample[1044:129397] [MC] Reading from public effective user
    settings.
2020-06-22 14:04:36.409284+0800 RtcSample[1044:129397] ---你好User1168
2020-06-22 14:04:36.409350+0800 RtcSample[1044:129397] joinChannel result: 16974081
2020-06-22 14:04:36.436340+0800 RtcSample[1044:129397] Focus mode not supported:
    AVCaptureFocusModeContinuousAutoFocus
2020-06-22 14:04:36.866295+0800 RtcSample[1044:129453] Metal API Validation Enabled
```

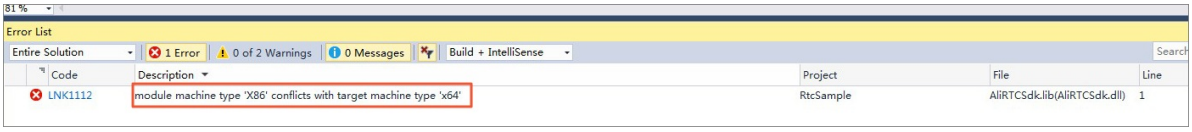
- 可能原因：channelID、userID或name命名不规范。
- 解决方案：根据规范，请您对channelID、userID、name重新命名。字符内容只允许[A-Za-z0-9_-]，长度限制64字节。

2.4. 集成Windows SDK时的常见问题

通过阅读本文，您可以了解集成Windows SDK时常见的问题及解决方法。

编译代码时报x64编译报错

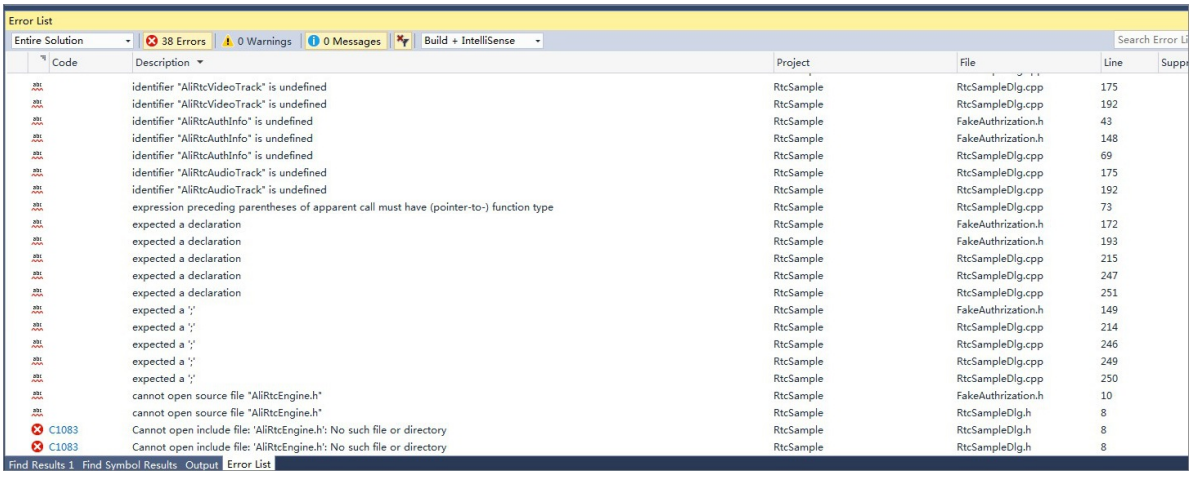
- 问题现象：编译代码时可能会出现以下错误：



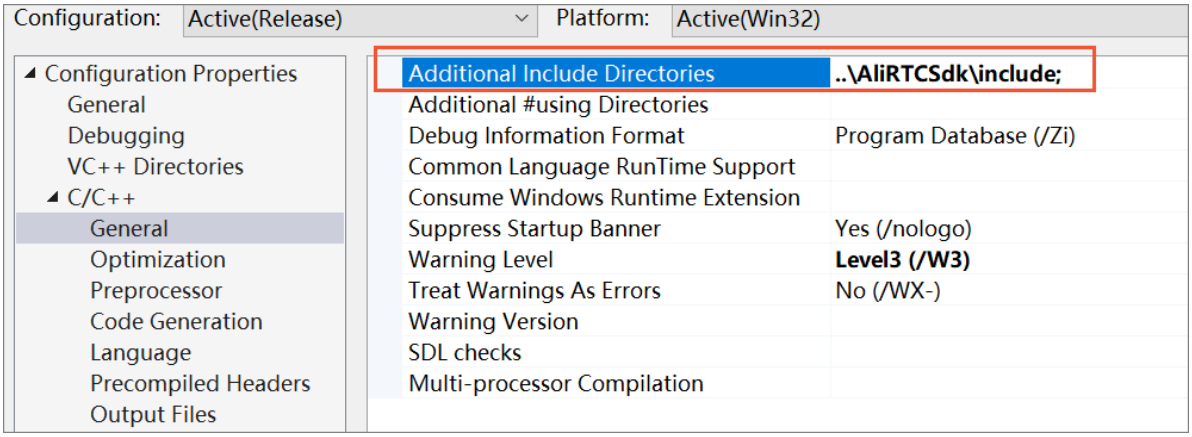
- 可能原因：使用64位编译。
- 解决方案：SDK目前只支持32位，请切换编译选项。

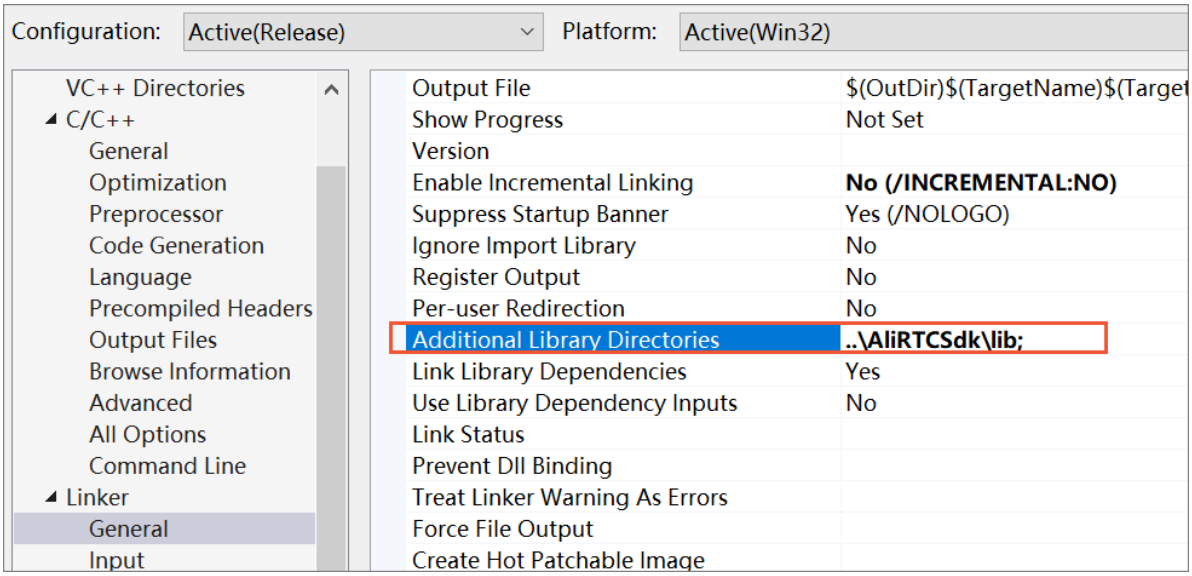
头文件或静态库路径设置错误

- 问题现象：编译代码时可能会出现以下错误：



- 可能原因：头文件或静态库路径未设置或设置错误。
- 解决方案：设置头文件和静态库路径。





动态库路径错误

- 问题现象：程序运行时可能会出现以下错误：

- 可能原因：缺少依赖的库文件。
- 解决方案：复制*AliRTCSdk.dll*及依赖的*ffmpeg.dll*到程序的执行路径下。

2.5. 集成Web SDK时的常见问题

通过阅读本文，您可以了解集成Web SDK时常见的问题及解决方法。

摄像头和麦克风无法使用

- 问题现象：程序运行时可能会出现以下错误：

  ```
  Error opening your camera and/or microphone: Only secure origins are allowed (see: https://goo.gl/Y0ZkNV).
  ```

- 解决方案：
  - AppServer和网页需要使用HTTPS协议。
  - 检测是否禁用或者占用摄像头和麦克风设备。

Web端和其它端无法互通

需要在其他端调用setH5CompatibleMode，设置兼容模式。

Android设备不能进行Web通信

检测Android设备是否支持H264协议。

> **说明** Android设备进行Web页面通信，需要安卓设备支持H264协议。

Web SDK无法内嵌App内

Web SDK不支持内嵌App内。

Web SDK是否支持小程序

Web SDK 暂不支持微信小程序、支付宝小程序。

无法显示本地摄像头

解决方案：

- 检查AppServer和网页是否使用HTTPS协议。
- 查看设备是否被禁用。
- 检查硬件是否可用。

SDK如果只使用音频，是否可以没有摄像头

如果只使用音频，可以没有摄像头。

2.6. Android SDK动态加载so文件方案

Android SDK提供so文件动态加载的功能，可以有效的减少安装包文件的大小。通过阅读本文，您可以了解在集成SDK时如何进行so文件的动态加载。

背景信息

传统安装包文件包含so文件，因此安装包文件通常会比较大，用户在下载使用时不太方便。使用so文件动态加载方案，即so文件不打包到apk里，在安装完并打开应用的时候通过后台下载so文件，将下载的so文件加载到应用中。阿里云RTC Android SDK提供动态加载方案所需的SDK分离包，将so文件和aar文件分离，开发者可以将aar文件打包进apk里，将较大的so文件放到云端进行下载，可以极大的减少安装包体积，方便用户下载使用。

注意事项

- 由于动态加载方案不能确保每次都会成功，如果so文件加载成功，可直接进入App；如果so文件没有加载成功，App会崩溃，建议执行降级操作或者重新加载so文件，成功之后才允许进入App。
- 建议在App空闲时下载so文件，提升使用体验。

集成SDK

方法一：Maven集成

1. 在根目录的build.gradle中添加Maven仓库地址：

```
allprojects {
    repositories {
        google()
        jcenter()
        //添加RTC需要的Maven地址
        maven {
            url "http://maven.aliyun.com/nexus/content/groups/public/"
        }
    }
}
```

2. 在项目的*app/build.gradle*文件中，添加如下行：

```
dependencies {
        ...
    //依赖的RTC SDK
    implementation 'com.aliyun.rtc:AliRTC:1.17.41.2102238'
}
```

> **说明** 此处Maven依赖的版本仅供参考，获取最新的Maven依赖，请参见精简版SDK。

3. 下载并解压so文件，下载地址，请参见精简版SDK。
4. 在代码中添加如下代码，加载so文件。

```
System.load("so文件绝对路径");
```

> **说明**
> - 下载后的文件解压后包含aar和so文件，请根据实际情况选择其中某一个so文件即可。
> - 此处不能使用 `System.loadLibrary()` 方式加载so文件，是因为 `System.loadLibrary()` 的参数为so文件名（不包含文件的扩展名），而且必须是在JVM属性 `java.library.path` 所指向的路径中，其他的路径不能识别，同时由于受到权限限制，下载的so文件没有办法保存在JVM属性 `java.library.path` 所指向的路径中，因此不能使用此方法；于此相反，使用System.load()时，System.load()的参数可以是任意路径，只要路径为so文件的绝对路径即可，因此进行so文件加载时，只能使用此方法。

5. 在*app/src/main/AndroidManifest.xml*文件中添加如下代码，获取相应的设备权限。

```
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS"/>
```

方法二：手动集成

1. 下载并解压Android SDK，下载地址，请参见精简版SDK。
2. 复制SDK文件AliRTCSdk.aar到App模块下的libs文件夹中。
3. 在代码中添加如下代码，加载so文件。

```
System.load("so文件绝对路径");
```

> **说明**
>
> - 下载后的文件解压后包含aar和so文件，请根据实际情况选择其中某一个so文件即可。
> - 此处不能使用 `System.loadLibrary()` 方式加载so文件，是因为 `System.loadLibrary()` 的参数为so文件名（不包含文件的扩展名），而且必须是在JVM属性 `java.library.path` 所指向的路径中，其他的路径不能识别，同时由于受到权限限制，下载的so文件没有办法保存在JVM属性 `java.library.path` 所指向的路径中，因此不能使用此方法；于此相反，使用System.load()时，System.load()的参数可以是任意路径，只要路径为so文件的绝对路径即可，因此进行so文件加载时，只能使用此方法。

4. 在*app/src/main/AndroidManifest.xml*文件中添加如下代码，获取相应的设备权限。

```
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS"/>
```

相关Demo

Android SDK动态加载so文件相关示例代码，请参见精简版SDK。

3.质量问题

3.1. 客户端入会失败常见原因

本章节为您介绍了客户端入会失败常见原因及解决办法。

加入频道所需参数如何获取

加入频道所需参数需要从AppServer获取，获取参数请参见服务端生成Token。

sessionID如何获取

RTS SDK 1.9以下版本中的sessionID字段可以置为空字符串，1.9及以上版本已经删除该字段。

常见典型错误码有哪些

错误码	说明
33620485	加入频道失败：Token无效，您可以重新生成频道鉴权令牌（Token），具体操作，请参见使用Token鉴权。
33620483	加入频道失败：频道已经失效（目前频道有效期为48小时），您可以重新生成频道鉴权令牌（Token），具体操作，请参见使用Token鉴权。

3.2. 客户端与服务端连通异常

本文为您介绍客户端与服务端连通异常的原因及解决办法。

问题现象

客户端与服务端连通异常。

可能原因

阿里云RTC需要使用UDP端口80、443、3478、50000。如果与服务端连通出现异常，可能是因为打开了防火墙或路由器策略限制导致。

解决方案

Windows和Linux平台下可以使用net cat命令测试UDP端口是否连通。

检查防火墙或路由器是否限制了UDP端口。

3.3. H5纯订阅模式媒体文件播放失败

本文为您介绍了H5纯订阅模式下浏览器页面不能自动播放媒体文件的原因及解决方法。

可能原因

浏览器为了防止网页在用户非自愿的情况下主动播放声音，对网页上的自动播放（Autoplay）功能做了限制：浏览器在没有用户交互操作之前不允许有声音的媒体播放。

受浏览器策略影响，Chrome 70+、Safari、Firefox等浏览器新版本都不支持带声音的媒体文件自动播放，需要您在网页上手动触发才能播放媒体文件。

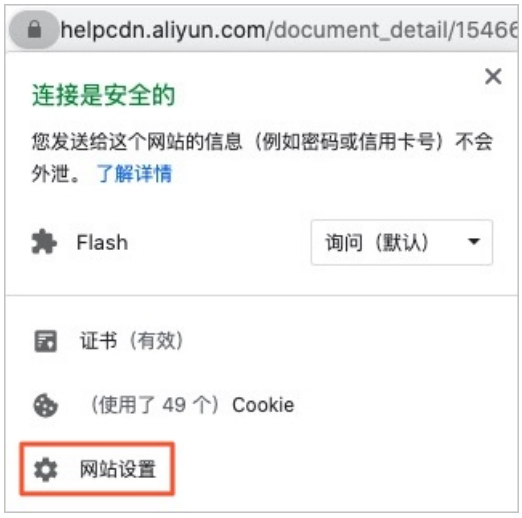
解决方案

- 在onError中获取到错误码10201时，会同时返回播放失败的 `userId` ，此时该用户的音频播放是静音的，在网页上手动触发事件（有用户交互）调用 `aliWebrtc.muteAllRemoteAudioPlaying(false)` 取消静音。
- 如果仅有几个人使用浏览器进行播放媒体文件，例如：视频直播网站的管理员。您可以通过设置浏览器来播放媒体文件，具体操作，请参见Chrome浏览器设置。

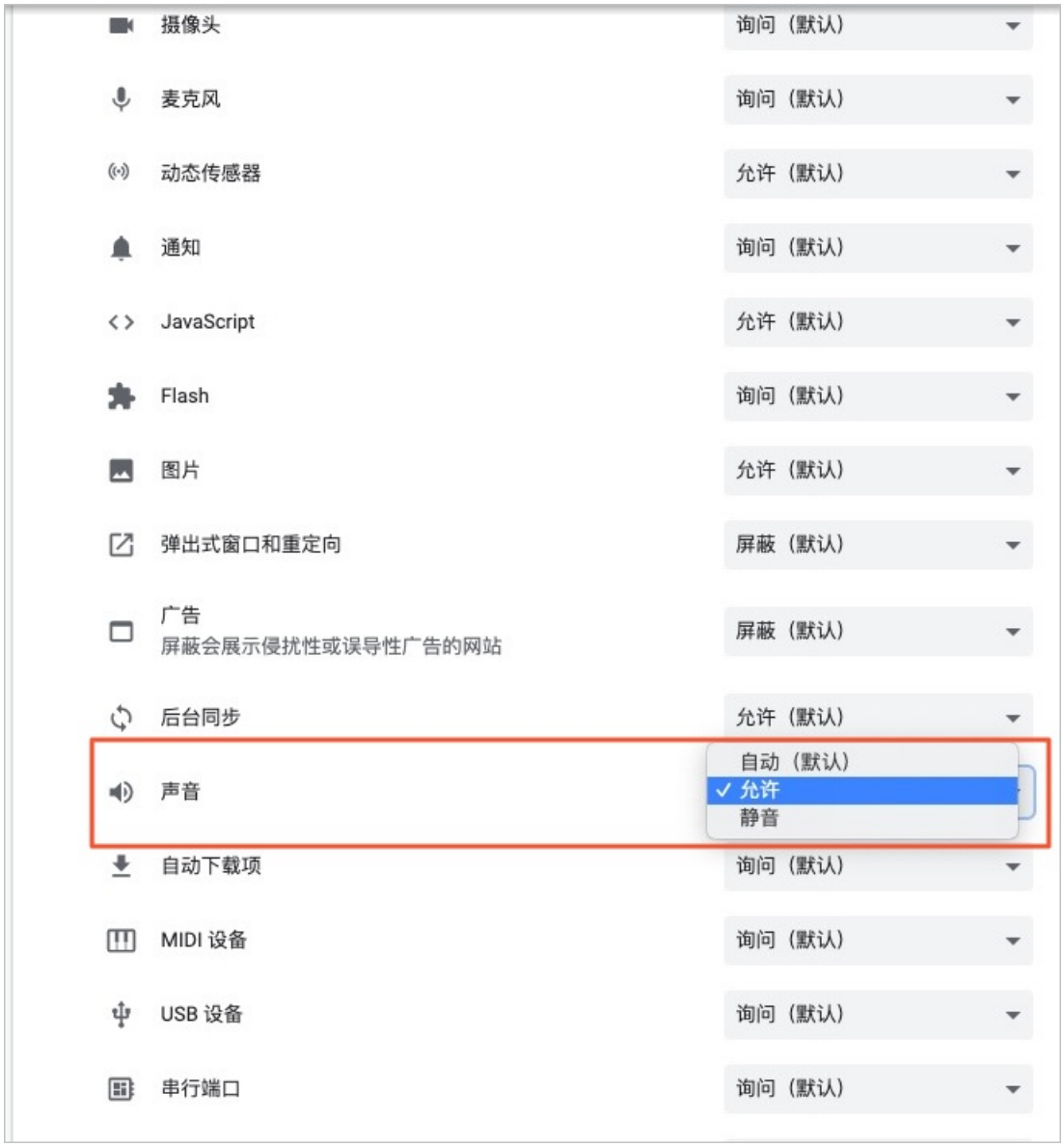
> 说明
> - 通过浏览器设置可以一次性解决某个域名下所有页面自动播放的问题，不再需要每次打开浏览器进行单独操作。
> - 本文涉及的浏览器设置仅供参考，具体操作请以浏览器实际版本为准。

Chrome浏览器设置

1. 单击浏览器地址栏左侧的🔒。
2. 单击**网站设置**。



3. 单击**声音**对应的下拉框，选择**允许**。



Safari浏览器设置

1. 在浏览器地址栏单击**此网站的设置...**。

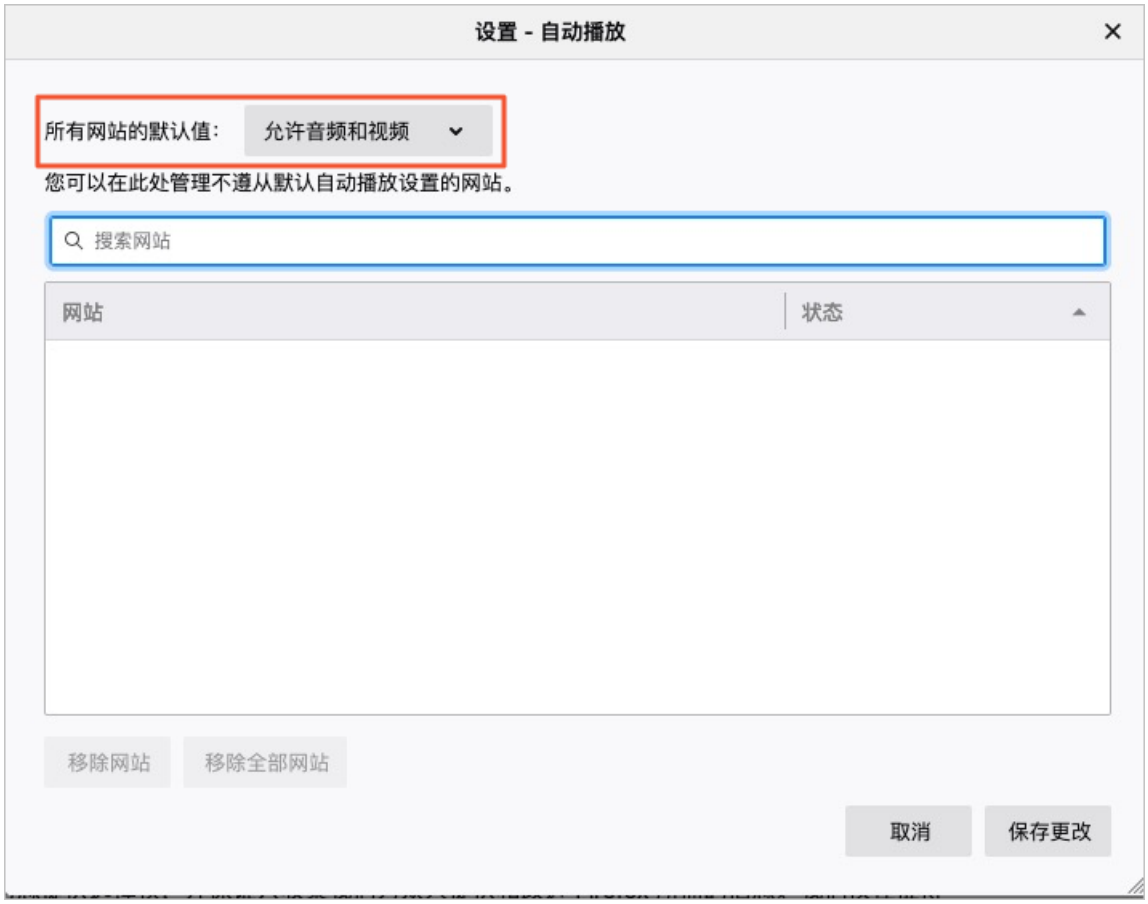
2. 将自动播放更改为**允许全部自动播放**。

Firefox浏览器设置

1. 单击浏览器地址栏左侧的。
2. 单击**保护设置**。

3. 单击**自动播放**对应的**设置**。

4. 单击下拉框，选择**允许音频和视频**。



4.SDK使用问题

4.1. 如何实现多模块共享AVAudioSession

iOS端RTC SDK和其他音频模块对AVAudioSession都有控制权，您需要根据实际情况分配AVAudioSession的控制权限。通过阅读本文，您可以了解怎样设置SDK对AVAudioSession的控制权限。

背景信息

默认情况下，RTC SDK和App对AVAudioSession都有控制权，为了保证RTC SDK通话功能的正常使用，RTC SDK会提高对AVAudioSession使用的优先级。然而在某些场景下，例如当需要暂停RTC SDK并且使用其他音频组件（音乐播放器、其他第三方音频组件等），此时需要限制RTC SDK对AVAudioSession的控制权限。

解决方案

RTC SDK为您提供以下接口方法设置SDK对AVAudioSession的控制权限，如果您调用了该方法限制了RTC SDK的控制权限，则需要App自行维护以保证SDK功能正常，App也可以随时使用此方法把管理权限交还给RTC SDK。

```
- (int)setAudioSessionOperationRestriction:(AliRtcAudioSessionOperationRestriction)restriction;
```

名称	类型	描述
restriction	AliRtcAudioSessionOperationRestriction	控制权限，默认值为AliAudioSessionOperationRestrictionNone。

4.2. SDK使用过程中如何减少耗时

使用SDK过程中如果存在不合理的配置或调用可能会增加耗时。通过阅读本文，您可以了解减少耗时的方法。

配置合理的自动订阅和自动推流

- 如果您的场景中需要入会后会默认推流，建议您采用自动推流模式。
- 如果您的场景中需要入会后默认订阅频道中的流，建议您采用自动订阅的模式。自动订阅模式在接收到远端用户回调之后，不需要再手动订阅，能更加节省时间。

RTC SDK默认自动推流和自动订阅，如果您没有其他特殊需求，无需调用推拉流接口进行设置。

场景中存在频繁切换频道或频繁入离会

- 避免频繁的创建和销毁SDK实例增加耗时，您可以通过一个SDK实例的入离会达到切换频道的效果。
- 根据实际需求，提前打开音频设备，节省每次入离会的耗时。例如：在语音直播间的场景中，您可以在创建SDK实例后调用一次startAudioPlayer接口提前打开播放设备，在销毁SDK实例前调用stopAudioPlayer接口关闭播放设备，中间切换频道和入离会均无需再额外调用。
- 根据实际需求，决定是否调用startAudioCapture和stopAudioCapture提前打开或关闭音频采集设备。

> **说明** 此处以Android端接口举例介绍，其它端接口类似，详情请参见RTC SDK接口文档。

4.3. Web端没有麦克风设备如何推视频流

通过阅读本文，您可以了解到当Web端没有麦克风时设备推视频流的方法。

1. 设置纯订阅模式跳过isSupport设备检测。

```
aliWebrtc.isSupport({isReceiveOnly: true}).then((re)=>{
    // 支持纯订阅模式
}).catch(err => {
    // 不支持纯订阅模式
})
```

2. 获取audiotrack，设置外部输入 `setExternalMediaTrack` 替换音频流。

```
//获取mediaStream
let mediaStream = video.captureStream(); // video为播放视频的media标签
//获取音频列表
let audiotracks = mediaStream && mediaStream.getAudioTracks()
//获取其中某一个audiotrack
let audiotrack = (audiotracks && audiotracks.length) ? audiotracks[0]
//置外部输入替换音频流
aliWebrtc.setExternalMediaTrack(audiotrack, 0);
```

> **说明**
> - 设置外部输入前需要先停止正在进行的推流。
> - 每次停止推流后都需要重新设置外部输入。

3. 正常推流（此时可以不推音频流，SDK内部识别出已经推了外部输入音频，因此会跳过设备检测）。

4.4. 通讯模式升级至互动模式说明

通过阅读本文，您可以了解到通讯模式和互动模式的区别以及从通讯模式升级到互动模式后，发布端和订阅端的操作说明。

背景信息

针对部分业务场景中，单一频道存在较多用户人数，但主要时间内只有一名主播推流，少量观众需要与主播连麦互动的情况，RTC SDK加入了新的频道模式（互动模式AliRtcInteractiveLive）。

通讯模式与互动模式的区别

通讯模式与互动模式的不同，主要体现在以下两点：

- 频道内最大支持人数不同，通信模式目前默认最多是支持50*50（即50个人同时推拉流）路流，互动模式默认最多支持1万人在线（麦上主播人数不建议超过16人，观众订阅流的总数不建议超过8000路）。
- 通信模式没有用户角色的区分，互动模式下增加了用户角色的设置功能，每位加入频道的用户都具备用户角色，分别为：
  - 互动角色（AliRtcChannelProfileInteractiveLive）：允许推流，适用于频道中的主播与连麦用户。

- 观看角色（AliRtcClientRolelive）：只能订阅拉流观看，不能推流，适用于普通观众用户。

同时，所有观众角色只能接受到互动角色的加入频道、推流、停止推流、离开频道等状态通知，不会收到其余观众用户的信令消息。

> 说明
> - 同一个频道里同时只能使用一种频道模式，频道模式由最早加入频道的用户设置，后续加入的用户也需要使用对应的频道模式，否则加入频道失败。例如，最早一个用户使用互动模式进入频道成功后，后续用户如果再使用通信模式加入频道将会失败。当频道内所有用户都离开2分钟后，该频道的频道模式会被重置，将由之后首个加入频道的用户设置的频道模式重新确定。
> - 由于前期SDK版本不支持频道模式设置，默认以通信模式加入频道。为避免出现上述因频道模式不匹配而加入失败，建议业务侧升级使用互动模式后，强制升级用户SDK版本，均升级完毕之后再启用，保证同一频道内用户频道模式统一。

通信模式升级为互动模式

从通信模式升级为互动模式时，发布端和订阅端需要进行以下操作：

发布端

- 主播在加入频道之前，先调用setChannelProfile接口设置频道模式为互动模式。
- 主播可以在加入频道前调用setClientRole接口，将用户角色设置为互动角色。主播入会之后自动推流，并且可以调整用户角色信息，当主播加入频道成功后保持自动推流。
- 主播也可以关闭自动推流模式，在加入频道后，切换角色为互动角色。当主播切换成功后会收到onUpdateRoleNotify回调通知，收到回调通知后可以配置需要推流媒体型，调用publish接口开始推流。

> 说明
> - 如果在加入频道成功前设置用户角色，默认初始角色即为设置角色，不会接收到角色变更回调。加入频道成功后再设置用户角色，设置为不同角色成功时会有回调通知。同时业务侧需要记录一下当前的角色参数值，重复设置当前角色不会收到回调。建议业务侧将角色切换和推拉流的逻辑区分开。
> - 用户必须确保在加入频道成功的前提下，进行切换角色操作和推流动作，否则无法切换和推流成功。

订阅端

- 用户在加入频道之前，先调用setChannelProfile接口设置频道模式为互动模式。
- 用户可以在加入频道前调用setClientRole接口，将用户角色设置为观看角色，关闭自动推流，保持自动订阅。
- 当连麦观众加入频道成功后，如果业务场景需要连麦，将设置用户角色为互动角色，设置成功后会收到onUpdateRoleNotify回调通知，收到回调通知后可以配置需要推流媒体型，调用publish接口开始推流；当连麦观众下麦时，关闭推流媒体并停止推流，将用户角色设置回观众角色与普通观众一样，保持订阅拉流即可。
- 对于普通观众，只需保持观看角色，订阅观看推流媒体，无需其他操作。

4.5. 视频双流规格配置说明

阿里云RTC服务支持双流（大小流）模式，您可以根据业务需求开启此功能。本文以iOS端为例介绍视频双流的实现方式及规格信息。

实现方式

在多用户视频通信时，您可以通过调用setVideoEncoderConfiguration接口将发布端的视频分辨率和码率适当调低，或调用publishLocalDualStream接口开启双流模式并通过调用setRemoteVideoStreamType接口设置订阅端默认订阅小流，以减少编解码的资源消耗和缓解下行带宽压力。更多信息，请参见设置视频属性。

> **说明** 开启双流模式后，RTC SDK在推送视频流时会额外再推送一路低分辨率、低码率的视频流。其中，原视频流称为大流，低分辨率、低码率的视频流称为小流。

双流规格默认对应关系

RTC SDK会根据大流视频的规格自动设置小流视频的默认规格，对应关系如下所示：

大流宽高比（宽：高）	小流		
	分辨率（宽 × 高）	码率（单位：Kbps）	帧率（单位：fps）
1：1	160 × 160	100	10
3：4	120 × 160	100	10
4：3	160 × 120	100	10
9：16	90 × 160	100	10
16：9	160 × 90	100	10
其他比例	大流宽高比中较大值对应小流分辨率中较大值160，较小值根据比例计算。	100	10

> **说明** 小流视频默认规格遵循以下规则：
>
> - 小流视频分辨率中较大值固定为160，并且和大流的宽高比一致。
> - 小流视频帧率固定为10 fps。
> - 小流视频分辨率的宽和高必须为偶数。
> - 小流视频码率固定为100 Kbps。

自定义小流规格

如果需要自定义小流视频规格，您可以在创建AliRtcEngine对象时设置入extras参数，例如：设置小流视频分辨率为320 × 180，帧率为5 fps，码率为140 Kbps，对应的extras参数设置如下：

```
{
  "user_specified_low_bitrate_stream_parameter": {
    "width": 320,
    "height": 180,
    "frameRate": 5,
    "bitRate": 140
  }
}
```

说明

- 设置小流视频分辨率时，宽取值范围为[64,640]，高取值范围为[64,480]。
- 设置小流视频帧率大于大流帧率时，小流帧率会自动调整为大流帧率；设置小流视频帧率为0或负数时，自动调整为10。
- 设置小流视频码率小于最小码率时，实际码率为设置的码率；设置小流视频码率大于最大码率时，实际码率为最大码率。

extras参数详细说明，请参见extras参数配置说明。

自定义大流规格

如果需要自定义大流视频规格，您可以根据环境实际带宽大小结合下表中视频分辨率、帧率和码率的对应关系进行设置。

分辨率（宽：高）	帧率（单位：fps）	码率（单位：Kbps）		
		最小值	推荐值	最大值（推荐值倍数）
160 × 120	5	10	30	× 3
	10	20	55	× 3
	15	30	80	× 3
	20	40	100	× 3
	30	60	130	× 3
120 × 120	5	10	25	× 3
	10	17	50	× 3
	15	25	70	× 3
	20	34	90	× 3
	30	50	115	× 3
	5	15	65	× 3
	10	30	110	× 3

分辨率（宽：高） 320 × 180	帧率（单位：fps）	码率（单位：Kbps）		
		最小值	推荐值	最大值（推荐值倍数）
	15	45	170	× 3
	20	60	220	× 3
	30	90	300	× 3
180 × 180	5	10	50	× 3
	10	17	70	× 3
	15	26	100	× 3
	20	34	130	× 3
	30	51	180	× 3
240 × 180	5	15	60	× 3
	10	30	90	× 3
	15	45	130	× 3
	20	60	165	× 3
	30	90	230	× 3
320 × 240	5	20	100	× 2
	10	40	170	× 2
	15	60	256	× 2
	20	80	320	× 2
	30	120	400	× 2
240 × 240	5	15	70	× 2
	10	30	100	× 2
	15	45	150	× 2
	20	60	200	× 2
	30	90	256	× 2
	5	26	100	× 2
	10	53	170	× 2

分辨率（宽：高） 424 × 240	帧率（单位：fps）	码率（单位：Kbps）		
		最小值	推荐值	最大值（推荐值倍数）
	15	79	260	× 2
	20	105	340	× 2
	30	158	430	× 2
640 × 360	5	83	200	× 2
	10	165	340	× 2
	15	248	512	× 2
	20	330	600	× 2
	30	495	700	× 2
360 × 360	5	30	120	× 2
	10	60	180	× 2
	15	90	260	× 2
	20	120	330	× 2
	30	180	400	× 2
480 × 360	5	40	150	× 2
	10	80	240	× 2
	15	120	350	× 2
	20	160	430	× 2
	30	240	512	× 2
640 × 480	5	110	200	× 2
	10	220	350	× 2
	15	330	512	× 2
	20	440	600	× 2
	30	660	700	× 2
	5	83	170	× 2
	10	165	260	× 2

分辨率（宽：高） 480 × 480	帧率（单位：fps）	码率（单位：Kbps）		
		最小值	推荐值	最大值（推荐值倍数）
	15	248	400	× 2
	20	330	470	× 2
	30	495	600	× 2
840 × 480	5	180	256	× 2
	10	360	512	× 2
	15	540	610	× 2
	20	720	800	× 2
	30	1080	930	× 2
1280 × 720	5	400	400	× 1.5
	10	800	800	× 1.5
	15	1200	1200	× 1.5
	20	1600	1600	× 1.5
	30	2400	2400	× 1.5
960 × 720	5	250	250	× 1.5
	10	500	500	× 1.5
	15	750	750	× 1.5
	20	1000	1000	× 1.5
	30	1500	1500	× 1.5
1920 × 1080	5	500	500	× 1.5
	10	1000	1000	× 1.5
	15	1500	1500	× 1.5
	20	2000	2000	× 1.5
	30	3000	3000	× 1.5
	5	800	800	× 1.5
	10	1600	1600	× 1.5

分辨率（宽：高） 2560 × 1440	帧率（单位：fps）	码率（单位：Kbps）		
		最小值	推荐值	最大值（推荐值倍数）
	15	2400	2400	× 1.5
	20	3200	3200	× 1.5
	30	4800	4800	× 1.5
3840 × 2160	5	1000	1000	× 1.5
	10	2000	2000	× 1.5
	15	3000	3000	× 1.5
	20	4000	4000	× 1.5
	30	6000	6000	× 1.5

说明

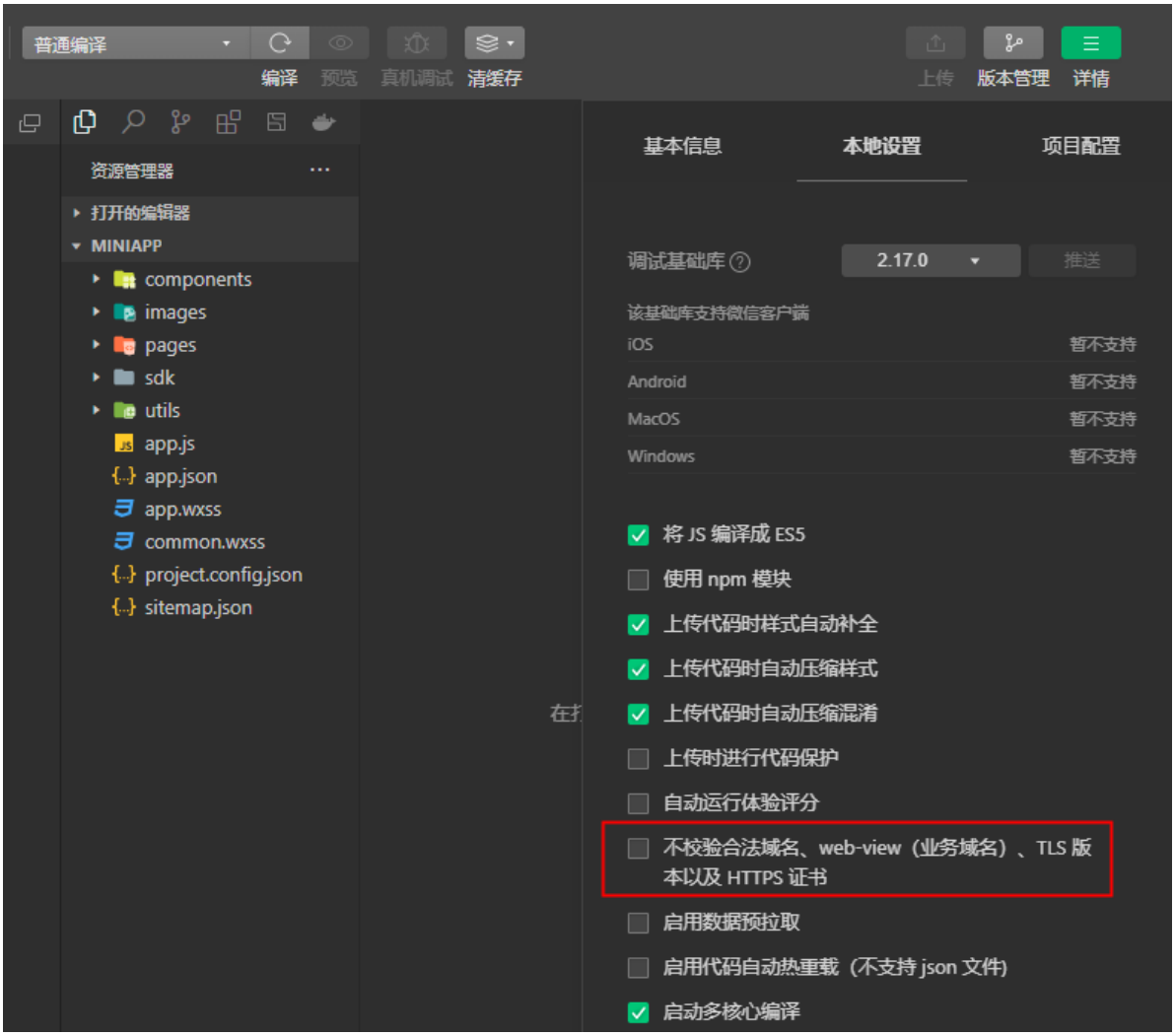
- 大流视频默认分辨率为640 × 480，帧率为15 fps，码率为512 Kbps。
- 如果大流为相机流，最高设置分辨率为1920 × 1080，帧率为30 fps。
- 如果大流为屏幕流，最高设置分辨率为3840 × 2160，帧率为30 fps。
- 建议您在设置视频规格时不要设置码率，RTC SDK会根据视频的分辨率和帧率设置合适的码率。

4.6. 小程序SDK常见问题

通过阅读本文，您可以了解使用小程序SDK时常见的问题及解决方法。

App闪退

- 问题现象：App闪退。
- 可能原因：启用其它WebSocket时没有开启验证HTTPS。
- 解决方案：启用其它WebSocket时，请确保**本地设置**不要选中**不校验合法域名、web-view（业务域名）、TLS 版本以及 HTTPS 证书**。



muteLocal接口未生效

- 问题现象：调用muteLocal接口时，功能没有生效。
- 可能原因：调用muteLocal接口后，没有设置live-pusher组件的 `enable-mic` 和 `enable-camera` 属性。
- 解决方案：调用muteLocal接口后，还需要设置live-pusher组件的 `enable-mic` 和 `enable-camera` 属性才能实现本地静音或关闭本地摄像头的功能，详情请参见live-pusher。

如何实现美颜、切换摄像头等功能

- 问题描述：如何实现美颜、切换摄像头等SDK没有提供的功能？
- 解决方案：实现美颜、切换摄像头等SDK没有提供的功能，需要参考live-pusher组件属性自行实现，详情请参见live-pusher。

音视频通话中如何保持屏幕长亮

- 问题描述：音视频通话中如何保持屏幕长亮？
- 解决方案：调用微信小程序接口 `wx.setKeepScreenOn({keepScreenOn: true})` 保持屏幕长亮，详情请参见setKeepScreenOn。

mute视频时音画不同步

- 问题现象：mute视频时音画不同步。
- 可能原因：mute视频会导致推流时某段时间不会发送音频数据，因此可能会造成流断开或者音画不同步。此问题属于小程序自身的缺陷，并非小程序SDK问题。

没有推流（采集黑帧）时如何设置默认显示图片

- 问题描述：没有推流（采集黑帧）时如何设置默认显示图片。
- 解决方案：设置小程序live-pusher组件中的 `waiting-image` 属性后，当推流端切换到后台，就可以推送静态图片，其他端拉流时会显示本端预设的 `waiting-image` 图片。

如何进行监听远端用户强制退出事件

- 问题描述：如何进行监听远端用户强制退出（没有调用停止推流直接离会）事件？
- 解决方案：当远端用户调用 `leave` 接口离会后，本地会收到 `leave` 事件的回调；当远端用户强制退出（没有调用 `leave` 接口）时，本地会在90s左右自动收到该用户的 `leave` 回调。

小程序可以在WebView网页中调用Web SDK的接口吗

- 问题描述：微信小程序可以在WebView网页中调用Web SDK的接口吗？
- 解决方案：微信小程序目前不支持在WebView网页中调用Web SDK的接口。要在微信小程序中实现实时音视频功能，需要在小程序项目中集成小程序SDK，详情请参见集成小程序SDK。

小程序SDK是否支持播放背景音乐

- 问题描述：在某些社交娱乐场景中，小程序应用场景需要播放背景音乐，阿里云微信小程序SDK是否支持播放背景音乐？
- 解决方案：小程序的背景音乐需要用调用微信小程序的原生API来实现，阿里云微信小程序SDK无法建立音频播放器。您可以通过调用 `wx.getBackgroundAudioManager()` 接口获取全局背景音频管理器，并调用 `BackgroundAudioManager` 类提供的接口管理背景音乐，详情请参见getBackgroundAudioManager和BackgroundAudioManager。

入会失败（错误码：100104）

- 问题现象：入会错误，提示错误码100104。
- 可能原因：没有配置域名白名单。
- 解决方案：配置服务器域名白名单。具体操作，请参见设置request合法域名和socket合法域名。

5.其他问题

5.1. H5端如何实现镜像及显示横屏录制的视频

Web SDK本身不提供镜像和横竖屏接口，需要您自己实现。通过阅读本文，您可以了解实现H5端镜像及显示横屏录制视频的方法。

H5端镜像

通过给video标签设置css，来实现镜像功能。

```
{
        transform: rotateY(180deg);
        -webkit-transform: rotateY(180deg);
        -moz-transform: rotateY(180deg);
}
```

显示横屏录制的视频

在video标签外增加一个div标签，并在div上设置css，来显示远端横屏录制的画面。

```
{
   transform: rotate(90deg);
   -webkit-transform: rotate(90deg);
   -moz-transform: rotate(90deg);
}
```

5.2. 旁路转推纯音频的配置说明

通过阅读本文，您可以了解通过调用StartMPUTask旁路转推纯音频的设置方法。

设置方法

纯音频模式和音视频模式的旁路转推参数配置类似，区别在于LayoutIds（布局），MediaEncode（编码选项）及TaskProfile（任务计费配置）参数的配置。

纯音频模式以上参数配置说明如下：

- LayoutIds：根据任务计费配置参数TaskProfile决定，TaskProfile设置为*Mixed_Audio*时不限制音频源数量。
- MediaEncode：只能设置为*0*。

更多参数说明信息，请参见StartMPUTask。

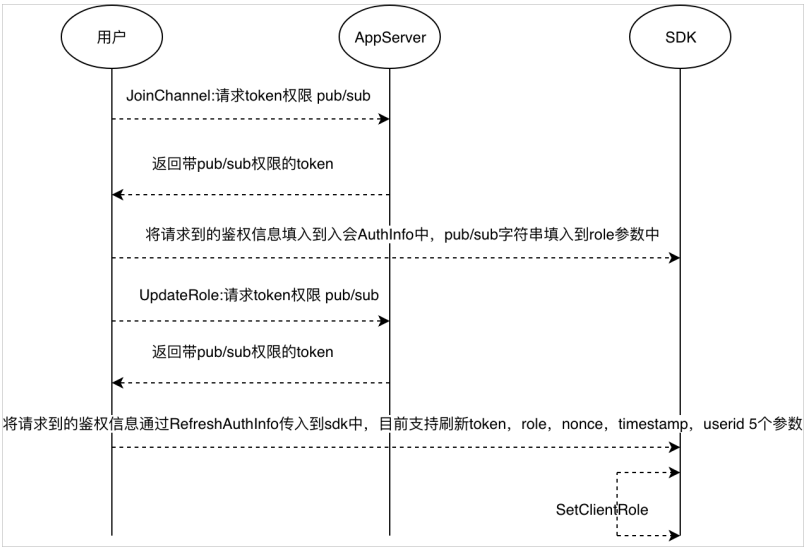
5.3. 如何进行角色鉴权

进行角色鉴权可以解决用户篡改RTC API调用逻辑，非法发布音视频流，防止没有推流权限的用户发布音视频流。通过阅读本文，您可以了解到角色鉴权的方法。

使用规则

- 角色鉴权目前仅在互动模式下生效。
- app_server计算token时，需要带上role参数的值，方法为在原来的字符串末尾增加role的值，再对整个字符串做加密。
- 主播角色入会时，必须授予鉴权角色为pub，否则无法入会。
- 观众角色入会时，可以授予角色为pub或者sub，授予pub权限的观众角色可以切换为主播角色。
- sub权限的观众角色，在切换为主播之前，需要向app_server请求pub角色的token，并且把新返回的token、role、nonce、timestamp、userid参数通过 `RefreshAuthInfo` 传入到SDK内，再调用 `setclientrole` 来切换成主播角色。

实现方法



操作步骤

1. 入会前，根据业务需求，向app_server请求相关信息，返回示例代码如下所示：

```
{
    "code": "0",
    "data": {
      "appid": "pdtk****",
      "userid": "4b73ca5c****",
      "nonce": "AK-1cdf3ef44a794eec9b208ad95b8d****",
      "timestamp": "1614738724",
      "roleToken": "sub",
      "token": "eeb8407849aeb52c1f9748f3652bccbed31b0a11e0d75dbaebb60fef42a2****",
      "turn": {
        "username": "4b73ca5c1226?appid=pdtk****&channel=12345&nonce=AK-1cdf3ef44a794ee
c9b208ad95b8d****&timestamp=1614738724",
        "password": "eeb8407849aeb52c1f9748f3652bccbed31b0a11e0d75dbaebb60fef42a2****"
      },
      "gslb": "https://pgslb.rtc.aliyuncs.com",
      "agent": "pagent.rtc.aliyuncs.com",
      "description": "Janus - Janus - 预发"
    },
    "server": "1614565924571"
}
```


将返回的 `roletoken` 的值填入入会参数role中。

2. 根据业务需求，主播角色需要请求 `tokenRole = pub` 的鉴权信息，为了避免观众非法破解获取推流的能力，观众角色可以请求 `tokenRole = sub` 的鉴权信息。
3. 入会后，如果需要切换角色，要再次向app_server请求相应的鉴权信息，获取到的信息通过 `RefreshAuthInfo` 接口传入到SDK内部。