实时音视频

SDK 参考

文档版本 01

发布日期 2024-10-24

华为云计算技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为云计算技术有限公司

地址： 贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址： https://www.huaweicloud.com/

目 录

1 使用前必读

1.1 主要功能

SparkRTC主要包含基本房间功能和跨房功能，各端主要功能框架，如图1-1所示。

说明：图1-1中仅展示各端的统一功能，独属功能详见各端SDK指导。

图 1-1 功能框架

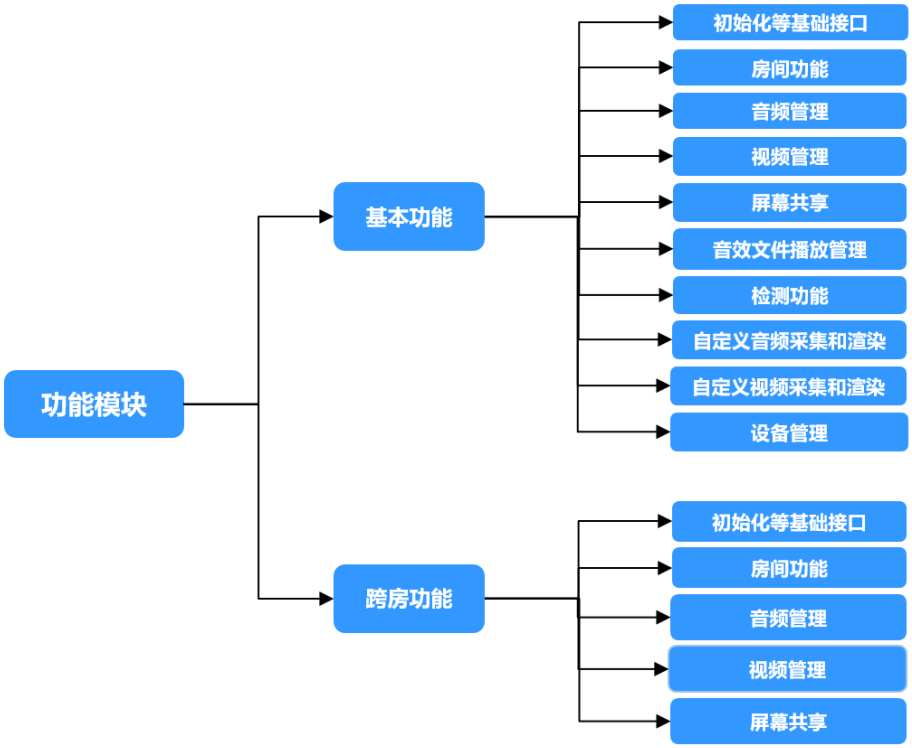


表 1-1 功能说明

类别	功能分类	功能说明
基本功能	初始化等基础接口	主要功能包括创建/销毁RTC引擎、设置日志保存位置等。
	房间功能	主要功能包括进入/离开房间操作、设置角色、创建跨房引擎等。
	音频管理	主要功能包括是否采集/发送本地音频流、是否接收远端音频流、调整录制/播放音量值、设置远端音频模式等。
	视频管理	主要功能包括创建本地/远端窗口视图和其他参数设置、是否接收远端视频流、镜像、摄像头等。
	屏幕共享	主要功能包括是否开始/停止订阅辅流、设置辅流渲染模式/角度等。
	音效文件播放管理	主要功能包括开始/停止/暂停/恢复播放音频或音效文件、音频/音效音量相关设置。
	检测功能	主要功能包括启动/关闭入会前网络检测。
	自定义音频采集和渲染	主要功能包括是否开启音频自采集、推送外部音频数据。
	自定义视频采集和渲染	主要功能包括是否开启视频自采集、推送外部视频数据、是否开启视频流自渲染。
	设备管理	主要功能包括切换摄像头、声音播放模式等。
跨房功能	跨房间连麦，指主播的媒体流可以同时转发进多个房间频道，实现主播跨频道与其他主播实时互动的场景。房间中的所有主播可以看见彼此，房间中的观众可以看到所有主播。同一时间最多只能同时跨4个房间，每个跨房房间的ID必须互不相同；同一时间只能以一个joiner角色加入某一个房间。如果本端在其他房间里的角色是joiner，则需要将本端在该房间内的player角色切换为joiner角色后再以joiner角色跨入其他房间。以player角色跨房后只能收流不能发流，以joiner角色跨房后既能收流也能发流。	

1.2 文档基本使用技巧

- **技巧1：基本使用逻辑说明**

  用时序图展示各端接口使用顺序，单击相应接口可以快速查看相关接口使用方法。

- **技巧2：接口总览说明**

  包括SparkRTC接口总体功能流程图和接口列表，根据功能分类可以快速查找具体功能单个接口，方便接口对接。

- **技巧3：单个接口使用须知**

  接口使用时注意使用的时机、参数说明。“注意”是强调每个接口使用的注意事项并带有调用该接口的相关回调，对接时需要仔细阅读。

1.3 常见问题分析解决办法

- **问题1：调用setVideoEncoderConfig接口时，为什么有些参数直接报参数设置错误?**

  设置分辨率时请使用华为SDK系统推荐的码表才能设置成功。

- **问题2：有哪些原因会导致跨房不成功?**
  - 可能原因1：同一时间不同房间最多只有一个Joiner角色才能跨房成功。
  - 可能原因2：同一时间最多只能跨4个房间，跨房对应的房间ID必须互不相同。

- **问题3：使用远端音频模式为HRTC_REMOTE_AUDIO_SUBSCRIBED时，如何设置才能默认听不到远端用户的声音?**

  HRTC_REMOTE_AUDIO_SUBSCRIBED为自主订阅，需要用户手动调用订阅。在加入房间（joinRoom）时调用带有HRTCJoinParam类的方法，创建该类实例后autoSubscribeAudio属性设置为false进入房间后则听不到远端用户的声音，需要手动调用muteRemoteAudio根据uid单个用户订阅才能听到声音。

- **问题3：为什么onVideoStats、onAudioStatus、onAuxiliaryStreamStatsNotify回调触发时程序崩溃?**

  回调函数的入参localStats和remoteStats指针有可能为空，需要先判断不为空再使用，否则可能引发空指针错误。

- **问题4：为什么本端听筒能听到自己的声音?**

  调用muteRemoteAudio时，参数设置为自己的uid就会发生此类情况。

- **问题5：setExternalAudioCapture（音频自采集）、setExternalVideoCapture（视频自采集）能在房间内开启吗?**

  不能，需要在加入房间前调用。

2 SDK 概述

华为云实时音视频服务软件开发工具包是对SparkRTC服务提供的REST API进行的封装，以简化用户的开发工作。用户直接调用SparkRTC SDK提供的接口函数即可实现使用SparkRTC服务业务能力的目的。

相关开发包请**提交工单**联系华为云技术客服获取。

同时，针对不同平台的SDK提供了集成和接口参考。详细介绍了SDK的集成操作、接口参数定义和代码示例。SparkRTC提供了以下主流平台SDK供开发者使用。

表 2-1 客户端 SDK

客户端	集成SDK	接口参考
Android	**Android SDK集成**	**Android SDK接口参考**
iOS	**iOS SDK集成**	**iOS SDK接口参考**
MAC	**MAC SDK集成**	**MAC SDK接口参考**
Windows	**Windows SDK集成**	**Windows SDK接口参考**
Web	**Web SDK集成**	**Web SDK接口参考**

软件包完整性校验

用户可对下载的SDK包进行完整性校验，判断下载过程中是否存在篡改和丢包现象。

详细操作如下所示：

步骤1 获取SDK包及其完整性校验sha256文件。

步骤2 打开本地命令提示符框，输入如下命令，在本地生成已下载SDK包的SHA256值。

其中，“D:\RtcSdk_Web_2.0.9.533.zip”为SDK包本地存放路径和SDK包名，请根据实际情况修改。

```
certutil -hashfile D:\RtcSdk_Web_2.0.9.533.zip SHA256
```

命令执行结果示例，如下所示：

```
SHA256 的 RtcSdk_Web_2.0.9.533.zip 哈希:
bca7141a498a17ee4eb1de208992c347daf65963140e614d0b3121ac62bca9be
CertUtil: -hashfile 命令成功完成。
```

步骤3 比对本地计算出的SDK包SHA256值和完整性校验sha256文件的SHA256值。

如果一致，说明下载过程中不存在篡改和丢包现象。

----结束

3 隐私声明

华为云SparkRTC SDK是由华为云计算有限公司（以下简称“我们”或“华为云”）面向华为云客户提供，方便用户接入实时音视频。华为云客户根据开发文档和用户指南，在其应用中集成**SparkRTC SDK**后，我们将通过被集成的**SparkRTC SDK**向华为云客户面向的最终用户（以下简称“您”或“用户”）提供相关服务，处理华为云客户的应用相关的数据，相关数据中可能包含您的个人信息。华为云非常重视您的个人信息和隐私保护，我们将会按照法律要求和业界成熟的安全标准，为您的个人信息提供安全保护措施。

我们将通过本声明向您说明我们如何收集、使用、披露、保护、存储及传输您的个人信息。

请注意，我们要求集成**华为云SparkRTC SDK**的所有华为云客户，**必须做到严格遵循法律法规、开发者协议去处理您的个人信息**。在接入、使用各开放能力前，华为云客户必须在其产品应用的隐私政策中，**向您告知其集成SDK处理个人信息的基本情况，并获取您的同意或取得其他合法性基础**。但我们无法控制华为云客户及其开发的应用如何处理华为云客户所控制的个人信息，也不对其行为负责。**我们建议您认真阅读华为云客户的应用相关用户协议及隐私政策**。在确认充分了解并同意，华为云客户如何处理您的个人信息后，再使用华为云客户的应用。

一、我们如何收集和使用您的个人信息

华为云仅会根据本声明以下所述目的和方式收集和使用您的个人信息。如果我们要将收集的个人信息用于本声明未载明的其他目的，我们会以合理的方式自行或通过华为云客户明确向您告知，并再次获取您的同意或取得其他合法性基础。如果SDK存在扩展功能，或收集和使用了您的可选个人信息，我们会在下文特别说明。

- **实时音视频的功能**

  为了向您提供实时音视频的功能，我们会处理您的音视频流，用于实时音视频的采集、编码、传输、播放等。这些内容数据不会被存储，仅在服务器内存中缓存，直播结束后自动清除。

- **实时音视频体验指标及问题定位的功能**

  为了向您提供实时音视频体验指标功能以及问题定位的功能，我们会处理您的IP地址、运营商信息、WiFi状态、浏览器信息、操作系统信息、设备型号等数据，用于对实时音视频的的分辨率、帧率、码率、卡顿率、丢包率、在线人数、视频质量等指标进行统计，以及用于协助您进行业务功能的问题定位和分析。您的上述数据将在中华人民共和国境内处理，这些内容数据的存储期限默认为90天。此为可选功能，可通过SDK接口打开关闭。

您的上述数据在中华人民共和国境内处理，我们不会将上述数据用作其他用途。您完全拥有上述数据所有权及控制权，**实时音视频SDK**仅接受委托处理您的数据。

二、设备权限调用

当您使用相应功能及服务时，我们会通过华为云客户的应用向系统申请您设备的相应权限。您可以在设备的设置功能或“隐私设置”中，查看权限状态，并选择开启或关闭部分或全部权限。华为云客户在集成使用相应开放能力时，可自行决定权限的调用范围，华为云客户向您说明权限调用的用途。您根据华为云客户的应用请求，开启任一权限，即代表授权我们处理相关个人信息来为您提供相应服务；一旦您关闭任一权限即代表您取消了授权，我们将不再基于对应权限继续处理相关个人信息，可能无法继续为您提供该权限所对应的功能。请注意，您关闭权限的决定，不会影响此前基于您授权所进行的个人信息处理活动的效力。当前华为云实时音视频SDK需要申请的权限如下：

权限	权限描述	使用目的
相机权限	使用摄像头	当您使用通话功能时，您可以选择开启该权限，用于给其他参与人传输您的视频画面
麦克风权限	使用麦克风	当您使用通话功能时，您可以选择开启该权限，用于给其他参与人传输您的语音
联网	访问网络	当您使用通话功能时，用于完成通话音视频文件的传输

三、对未成年人的保护

- 在您开始使用本服务时，须承诺您是成年人。若您是未成年人，须您的父母或监护人同意您使用本服务及相关服务条款。
- 如果未成年人在未经父母或监护人同意的情况下，向我们提供了个人信息，父母或监护人可以联系我们，停止收集、使用或披露其个人信息。
- 如果我们发现在未事先获得可证实的父母或监护人同意的情况下，收集了未成年人的个人信息，会设法尽快删除相关数据。
- 如果华为云客户使用本服务用于教育用途，且您的最终用户可能是未成年人，请确保您的最终用户使用本服务前已获得其父母或监护人的明确同意。

四、管理您的个人信息

华为云非常尊重您对个人信息的关注，我们将遵照相关法律法规的要求，协调、支持并保障您行使访问、复制、更正或删除个人信息操作的主体权利。

由于您是通过华为云客户的应用使用**华为云SparkRTC SDK**和服务，如果您希望访问、复制或更正与**华为云SparkRTC SDK**的个人信息，您应通过华为云客户的应用提供的路径实现您的个人信息主体权利。

为保障您访问、复制、更正或删除个人信息的权利实现，我们在与华为云客户的协议中，明确要求华为云客户承诺根据法律法规要求，向您提供便捷的权利实现方式。同时，我们的开放能力也向华为云客户提供了相关的接口，支持华为云客户通过接口调用方式来执行您关于个人信息的访问、复制、更正、删除的权利请求。您也可以通过本声明中“**如何联系我们**”所述联系方式与我们取得联系，我们将尽力协调、支持并保障您的上述权利实现。

当您直接向我们提出个人信息主体权利时，为了保障您的数据安全和其他合法权益，我们可能会对您的身份进行验证并要求您提供验证身份所必要的个人信息，同时我们

也可能会向华为云客户提供收集的身份验证信息以核实您的身份。在验证确认您的身份后，我们会根据法律法规要求及时响应您的相关请求。

如您对您的数据主体权利有进一步要求或存在任何疑问、意见或建议，可通过本声明中“**如何联系我们**”所述方式与我们取得联系，并行使您的相关权利。

五、信息存储地点及期限

- **存储地**

  上述信息将会传输并保存至中华人民共和国境内的服务器。

- **存储期限**

  音视频数据在拉流结束后，会立即删除。其他数据会在使用结束90天后删除。

  我们仅在实现本声明所述目的所必需的时间内，保留您的个人信息。并在超出保留时间后，删除或匿名化处理您的个人信息，除非法律法规另有要求。

六、如何联系我们

我们设立了个人信息保护专职部门。当您有任何疑问、建议、投诉、请求，请通过访问**隐私问题页面**与我们联系。我们将尽快处理您提交的问题，并在15个工作日或法律法规规定的期限内，答复您的问题。 如果您对我们的回复不满意，特别是认为我们的个人信息处理行为损害了您的合法权益，您还可以向有管辖权的个人信息保护机构或其他监管部门进行投诉或举报。 一般情况下，我们会尽最大努力响应和处理您的请求。结合您的请求或问题的复杂程度，我们可能会有所延迟，但我们会告知您延迟的理由。

4 合规使用指南

华为云SparkRTC SDK开发者合规指南

《 中华人民共和国个人信息保护法 》自2021年11月1日起正式施行后，监管部门、各行业参与方和终端消费者越来越关注用户的隐私保护问题。为了有效治理App、SDK违规收集使用个人信息的现象，监管部门也陆续出台相关标准规范。

您作为开发者为最终用户提供服务，知悉并确认将遵守适用的法律法规和相关的标准规范，履行个人信息保护义务，并遵循合法、正当、必要和诚信的原则处理用户个人信息，包括但不限于《 中华人民共和国个人信息保护法 》、《 中华人民共和国网络安全法 》、《 中华人民共和国数据安全法 》以及其他适用的法律法规和相关的标准规范。

此文档用于帮助您更好地了解【 华为云实时音视频SDK 】并合规的使用【 华为云实时音视频SDK 】服务，仅适用于开发者的业务区域为中国大陆地区的场景。

一、基本要求

您的产品及服务需要尊重用户隐私，遵守国家的数据保护法律和法规。禁止参与任何干扰、干涉、损害、未授权访问任何终端设备、服务器、网络的活动。

1. 隐私政策要求

您需根据法律要求以自身名义发布隐私政策，并就个人信息的处理行为获取用户同意或取得其他合法性基础。隐私政策的要求包括不限于如下：

- 有独立文本，不能作为用户协议的一部分。
- App首次运行收集处理个人信息前需要以醒目方式提示用户阅读隐私政策。隐私政策需方便用户查看，例如用户在App主功能界面中通过4次以内的点击或滑动操作可访问。
- 描述语言需要清晰通俗，符合通用语言习惯，避免使用有歧义的语言。
- 隐私政策内容要包含产品及服务收集个人信息的目的、方式和范围，个人信息处理者的名称和联系方式等。
- 您的产品及服务如涉及向第三方共享个人信息或集成了第三方的SDK时，需要在隐私政策中向用户进行披露和说明，获取用户的授权或同意。

2. 处理个人信息要求

您的产品及服务在处理用户个人信息时，需要遵守的要求包括不限于如下：

- 处理个人信息需要基于使用目的所必需，满足最小化原则。

- 实际收集和处理的个人信息范围、使用目的需要与隐私政策的范围保持一致。
- 收集个人信息的频率需与隐私政策保持一致，禁止超频次收集个人信息。
- 有明确的个人信息到期删除机制，个人信息的存留期与隐私政策保持一致，到期按时删除个人信息或对个人信息进行匿名化处理。
- 如涉及处理不满十四周岁未成年人个人信息前，应取得未成年人的父母或其他监护人的同意。
- 如涉及处理个人信息用于个性化推荐功能或大数据分析业务的，应告知并取得最终用户的授权同意情况下方可开展相关业务功能。
- 如涉及处理敏感个人信息前，应取得最终用户的单独同意。
- 如涉及跨境传输个人信息，需要按照国家网信部门会同国务院有关部门制定的办法和相关标准进行安全评估，并符合其要求。同时您还取得最终用户的单独同意。
- 支持用户方便的行使数据主体权利，例如查阅、复制、更正、删除个人信息等权利。

二、声明SDK处理的个人信息

在您接入、使用【 华为云实时音视频SDK 】服务前，我们要求您在隐私政策中向用户告知我们SDK的名称、SDK提供方名称、收集个人信息类型、使用目的、隐私政策链接，并获取用户的同意或取得其他合法性基础。您可以参考如下方式提供条款内容：

1. 以文字方式向用户告知

第三方SDK名称：华为云实时音视频SDK

第三方公司名称：华为云计算技术有限公司

收集个人信息类型：

音视频流、IP地址、运营商信息、WiFi状态、浏览器信息、操作系统信息、设备型号。

使用目的：为您提供实时音视频服务，向您提供实时音视频体验指标功能，以及方便问题定位。

隐私政策链接：隐私声明

2. 以表格方式向用户告知

第三方SDK名称	第三方公司名称	收集个人信息类型	使用目的	隐私政策链接
华为云实时音视频SDK	华为云计算技术有限公司	音视频流、IP地址、运营商信息、WiFi状态、浏览器信息、操作系统信息、设备型号	为您提供音视频直播服务，向您提供实时音视频体验指标功能，以及方便问题定位。	隐私声明

三、权限使用要求

我们SDK在提供服务时会最小化的使用系统权限，您需要根据实际使用的功能申请对应的系统权限并向用户告知征得其同意。

权限	权限描述	使用目的
相机权限	使用摄像头	当您使用通话功能时，您可以选择开启该权限，用于给其他参与人传输您的视频画面
麦克风权限	使用麦克风	当您使用通话功能时，您可以选择开启该权限，用于给其他参与人传输您的语音
联网	访问网络	当您使用通话功能时，用于完成通话音视频文件的传输

四、延迟初始化要求

为了避免您的应用在未获取用户的同意前SDK提前处理用户的个人信息。我们提供了SDK初始化的接口create，用于创建实时音视频引擎，请保证您的应用获取用户同意后才能调用此接口初始化SDK。

五、最小化使用功能要求

我们的SDK针对扩展功能和可选的个人信息的处理提供了配置能力，您可以基于业务诉求选择开启或关闭相关功能，实时音视频SDK提供开关上报日志的方法。

1. enableStats接口中enabled参数，true表示打开，false表示关闭，用于控制是否打点上报日志。

2. create接口中logEnable设置为false，则关闭日志输出；设置为true，则会打印输出日志。

5 Android SDK

5.1 开发前准备

前提条件

已**提交工单**获取SDK包。

环境要求

OHOS SDK需要集成到APP工程中，建议您在如下推荐环境中进行集成开发。

- 准备DevEco，推荐使用4.0.0(10) SDK及以上。
- 准备Android运行环境：API 21、Android 5.0以上设备。
- 支持的终端CPU架构：armeabi-v7a、arm64-v8a。

说明

手机的CPU架构可通过以下方式查询。

手机开启USB调试，连接上电脑，然后打开Windows操作系统中的cmd程序，输入如下命令：

```
adb shell getprop ro.product.cpu.abi
```

SDK 集成

步骤1 解压Android SDK包。

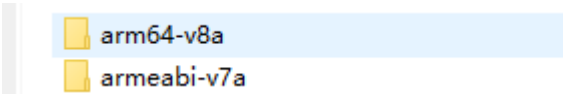
步骤2 将Android SDK包中的“hwRtcSdk.aar”等aar文件，导入Android Studio工程的libs文件夹下。

步骤3 在“/app/build.gradle”文件中设置依赖本地aar。

```
// 依赖本地aar
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar', '*.aar'])
}
```

步骤4 将如**图5-1**所示的包含so库的两个文件夹导入到jniLibs文件夹中。

图 5-1 so 库目录



步骤5 在“/app/build.gradle”文件中设置so库的存放路径。

```
sourceSets {
    main {
        jniLibs.srcDirs = ['src/main/jniLibs']
    }
}
```

步骤6 在“app/src/main/res/values/strings.xml”文件中配置appId。其中，appId请参考**应用管理**获取。

```
<string name="setting_appId_title" translatable="false">appId</string>
```

步骤7 在“/app/src/main/AndroidManifest.xml”文件中配置App权限。

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

步骤8 单击“Sync Project With Gradle Files”，同步项目文件，完成SDK集成。

----结束

5.2 SDK 使用

步骤1 创建引擎。

AppId获取方法请参见**创建应用**。

```
HRTCEngineConfig config = new HRTCEngineConfig();
config.setAppId(appId); // AppId需在控制台中创建应用后获取
config.setCountryCode(countryCode); // 可以根据Grs国家码对照表传值，建议传"CN"
config.setContext(getApplicationContext()); // 上下文，请传入Application Context
config.setDomain(webSocketFalvor); // 该字段已废弃，不需要再传值
config.setMuteAudioRoute(isMuteAudioRoute);
config.setLogEnable(true);
config.setLogSize(logSize);
config.setLogLevel(logLevel);
config.setLogPath(logPath); // logPath为目录，非文件
mHwRtcEngine = HRTCEngine.create(config, mHwHandler); // mHwHandler继承自
IHRTCEngineEventHandler，用于监听各种回调事件
```

设置本地窗口。

```
    SurfaceView surface = mHwRtcEngine.createRenderer(getApplicationContext()); // 不可使用new
SurfaceView(context)创建
    mHwRtcEngine.setupLocalView(surface,
HRTCEnums.HRTCVideoDisplayMode.HRTC_VIDEO_DISPLAY_MODE_HIDDEN);
```

加入房间。

```
HRTCJoinParam joinParam = new HRTCJoinParam();
joinParam.setUserId(mUserId); // userId用于标识同一房间的不同用户
joinParam.setUserName(mUserName); // 用户昵称，如无特殊需求，保持和userId一致即可
joinParam.setRole( HRTCJoinParam.HRTCRoleType.HRTC_ROLE_TYPE_JOINER);
joinParam.setRoomId(roomid);
joinParam.setScenario(1);
joinParam.setOptionalInfo(optionInfo);
```

```
joinParam.setAuthorization(signature);
joinParam.setCtime(ctime);
joinParam.setAutoSubscribeVideo(false);
joinParam.setAutoSubscribeAudio(true);
mHwRtcEngine.joinRoom(joinParam );
```

joinParam：入会参数，包含用户ID、用户名、房间号、认证信息、ctime、是否自动订阅音频和视频、SFU类型、场景和用户角色，具体请参见**HRTCJoinParam**。

步骤2 监听远端用户加入房间，并设置远端窗口。

```
@Override
public void onRemoteUserOnline(String roomId, final String userId, String nickname) {
    if (userId.equals(mUserId)||mRole == HRTC_ROLE_TYPE_JOINER.ordinal()) {
        return;
    }
    Log.e(TAG, "onRemoteUserOnline userId: " + userId);
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            Toast.makeText(LiveActivity.this, userId + "加入了房间", Toast.LENGTH_SHORT).show();
            SurfaceView surface = mHwRtcEngine.createRenderer(getApplicationContext());
            mHwRtcEngine.startRemoteStreamView(userId,surface,
HRTCEnums.HRTCStreamType.HRTC_STREAM_TYPE_HD);

mHwRtcEngine.updateRemoteRenderMode(userId,HRTCEnums.HRTCVideoDisplayMode.HRTC_VIDEO_DISPL
AY_MODE_Fit);
            // 将userId对应surface添加到布局中
        }
    });
}
```

步骤3 监听远端用户离开房间，并删除远端窗口。

```
@Override
public void onRemoteUserOffline(String roomId, final String userId, int reason) {
    Log.i(TAG, "HwRtcDemo onRemoteUserOffline roomId:" + roomId + ", userId:" + userId + ", reason:" +
reason);
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            Log.e(TAG, "HwRtcDemo run removeRemoteUser! ");
            Toast.makeText(LiveActivity.this, userId + "离开了房间", Toast.LENGTH_SHORT).show();

            mHwRtcEngine.stopRemoteStreamView(userId);
            // 将userId对应的surface从布局中移除
        }
    });
}
```

步骤4 离开房间。

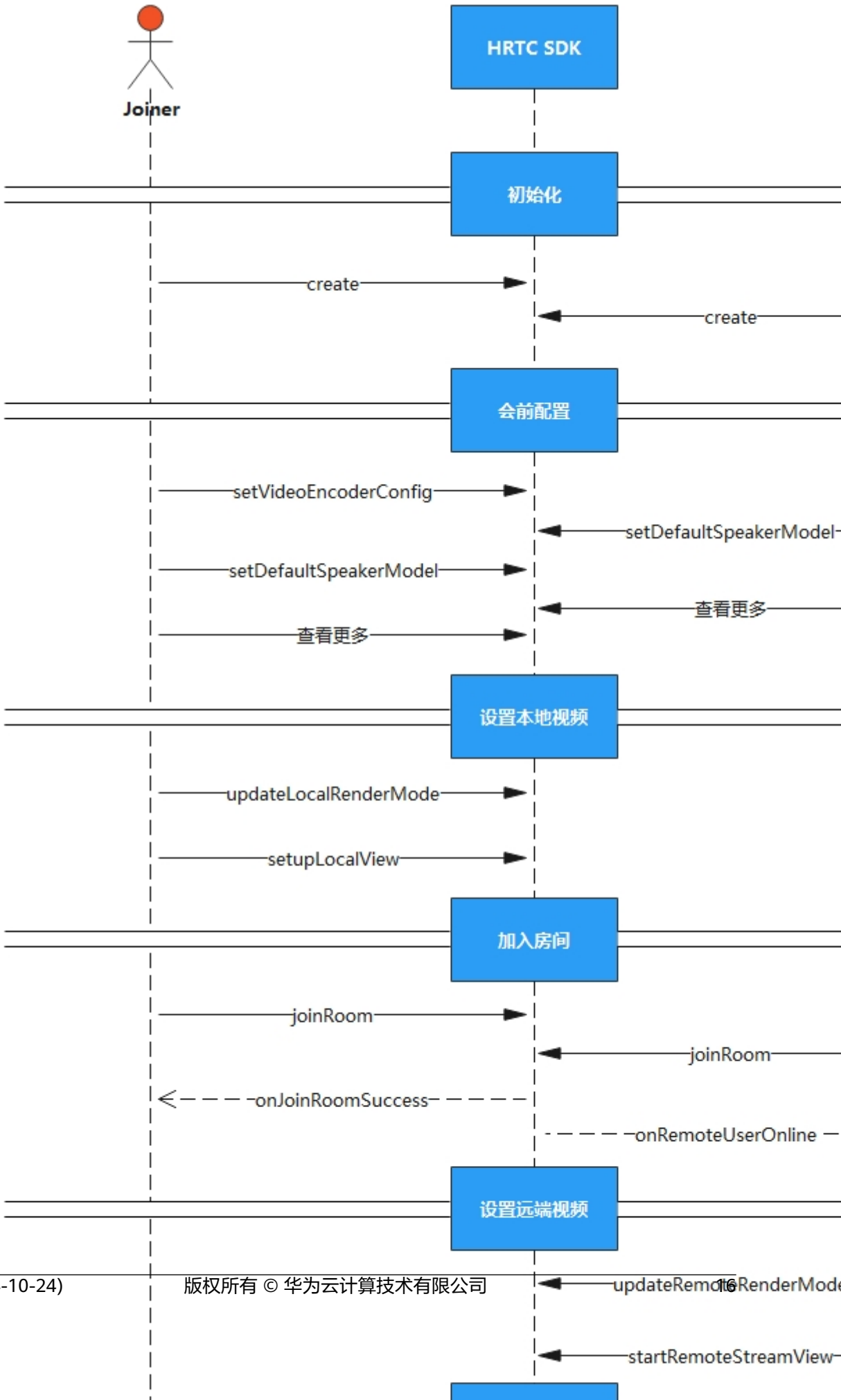
```
mHwRtcEngine.leaveRoom();
finish();
```

步骤5 销毁引擎

```
HRTCEngine.destroy();
```

----结束

5.3 基本使用逻辑



1. 创建新的项目工程，导入SDK后，需要先创建引擎。
2. 您可以在入会前进行视频编码、声音播放模式等参数的配置。
3. 设置本地视图。
4. 用户加入房间后，将通过回调的方式通知房间内的其他用户，收到其他用户加入的回调后，可以为其设置远端视图。
5. 在会中，也可以进行切换摄像头等参数的配置。
6. 用户离开房间后，需销毁对应资源。

说明

在时序图中，单击相应接口名称可快速跳转到相应接口位置查看其使用方法。

5.4 接口参考

5.4.1 HRtcEngine

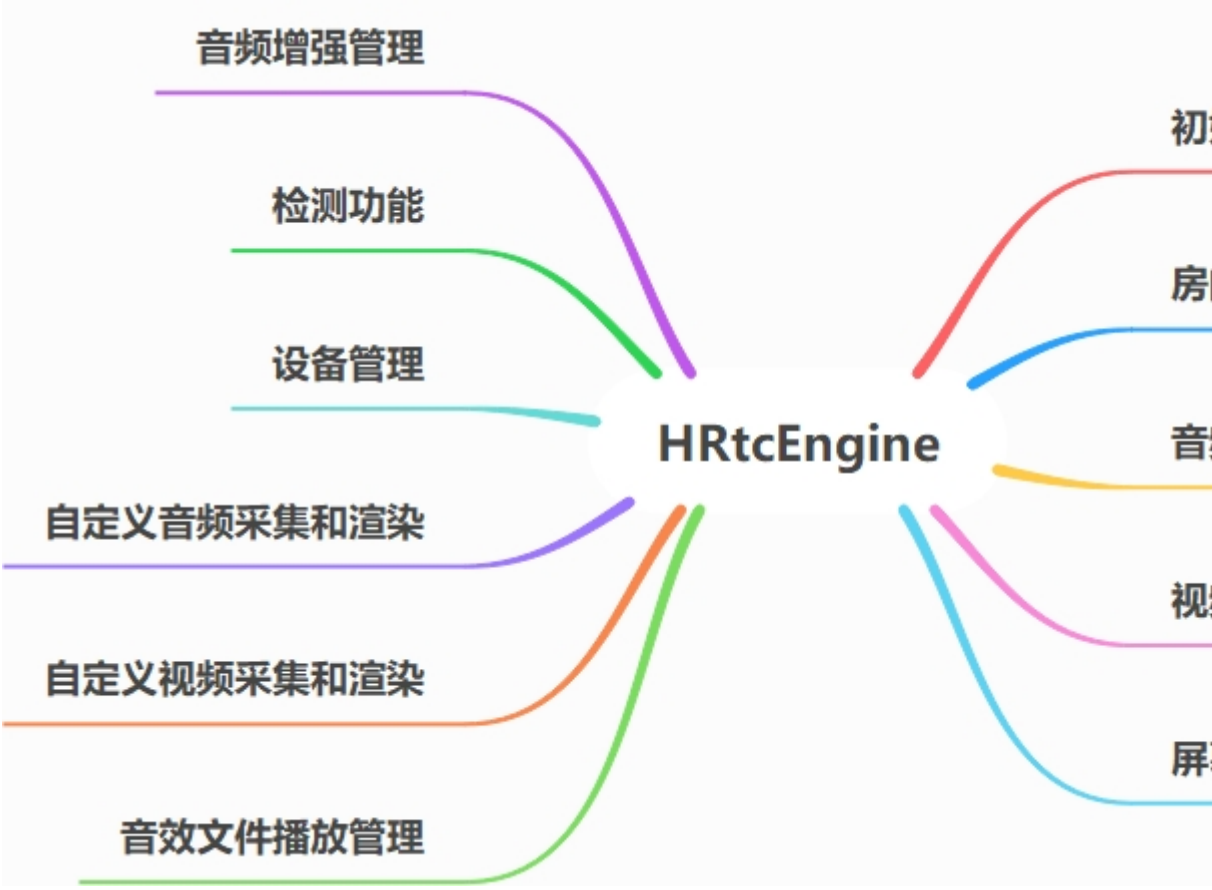
5.4.1.1 接口总览

本章节介绍了Android SDK的HRtcEngine接口详情。

HRtcEngine按照其功能可分类为：初始化等基础接口、房间功能、视频管理、屏幕共享、音频管理、音效文件播放管理、自定义视频采集和渲染、自定义音频采集和渲染、设备管理、检测功能、音频增强管理。

说明

单击下图中相应接口名称，可快速跳转到相应接口位置查看其使用方法。



初始化等基础接口

表 5-1 初始化等基础接口

接口	描述
create	创建SparkRTC引擎实例
destroy	销毁SparkRTC引擎
logUpload	上传日志
disableRejoinRoom	设置是否禁用房间重连功能。
enableStats	打点功能开关
getVersion	获取当前SDK版本号
setEncryption	设置端到端加密模式
setAccessResourceType	设置接入环境
setNetworkBandwidth	设置网络带宽限制

房间功能

表 5-2 房间功能接口

接口	描述
joinRoom	加入房间
leaveRoom	离开房间
changeUserRole	设置用户角色
renewAuthorization	更新鉴权签名
changeUserName	更新用户昵称
createConnection	创建跨房（HRTCConnection）对象
addMultiRoomMediaRelay	添加单个跨房
removeMultiRoomMediaRelay	删除单个跨房
stopMultiRoomMediaRelay	停止所有跨房

音频管理

表 5-3 音频管理接口

接口	描述
muteLocalAudio	设置是否发送本地音频流
muteRemoteAudio	设置是否接收对应远端用户的音频流
muteAllRemoteAudio	设置是否接收所有远端用户的音频流
enableLocalAudioStream	设置是否开启音频采集
adjustRecordingVolume	调整录制音量
adjustPlaybackVolume	调整播放音量
adjustPlaybackVolume	调整单个用户播放音量
sendAudioSeiMsg	发送音频SEI
setAudioFrameRecordParameters	设置音频采集回调参数
setAudioConfig	设置音频场景

视频管理

表 5-4 视频管理接口

接口	描述
createRenderer	创建渲染视图
updateLocalRenderMode	设置本地窗口显示模式，镜像模式
setupLocalView	设置本地窗口
pushLocalVideo	设置是否发送本地视频流
setVideoEncoderConfig	设置视频编码参数
setNonStandardVideoEncoder	设置非标视频编码参数
setRemoteVideoAdjustResolution	设置订阅视频流的分辨率自适应
startLocalPreview	开始本地预览
stopLocalPreview	停止本地预览
startRemoteStreamView	开始订阅远端视频流
stopRemoteStreamView	停止订阅远端视频流
setupRemoteView	设置远端窗口视图
updateRemoteRenderMode	设置远端窗口渲染模式，镜像模式
pullRemoteVideo	设置是否接收对应远端用户的视频流
pullAllRemoteVideo	设置是否接收所有远端用户的视频流
setVideoEncoderMirror	设置视频编码镜像模式
enableLocalVideo	设置是否开启摄像头采集视频
enableVideoSuperResolution	设置是否开启视频超分
enableSmallVideoStream	开启并设置小流编码参数
setPriorRemoteVideoStreamType	设置默认订阅的视频流类型（大流还是小流）
setRemoteVideoStreamType	设置当前订阅的视频流类型
startPublishStream	开始旁路推流
updateTransCoding	更新旁路推流
stopPublishStream	停止旁路推流
startAllRemoteView	批量设置远端流视图

屏幕共享

表 5-5 屏幕共享接口

接口	描述
startScreenShare	开启屏幕共享
stopScreenShare	关闭屏幕共享

辅流管理

表 5-6 辅流管理接口

接口	描述
setAuxiliaryVideoEncodeSmooth	设置是否开启辅流的流畅度优先。
startRemoteAuxiliaryStreamView	开始订阅辅流
stopRemoteAuxiliaryStreamView	停止订阅辅流
setRemoteAuxiliaryStreamViewRotation	设置辅流角度
updateRemoteAuxiliaryStreamRenderMode	设置辅流渲染模式，镜像模式
setAuxiliaryVideoEncoderConfig	设置辅流编码参数
setAuxExternalVideoCapture	设置是否开启视频辅流外部采集

音效文件播放管理

表 5-7 音效文件播放管理接口

接口	描述
startAudioFile	开始播放音频文件
stopAudioFile	停止播放音频文件
pauseAudioFile	暂停播放音频文件
resumeAudioFile	恢复播放音频文件
isPlayMixMyself	是否只有本地可以听到混音
isMixWithMicrophone	是否需要替代采集
adjustAudioFileVolume	调整本地和远端音频播放的音量
adjustAudioFilePlayoutVolume	调整本地音频播放的音量

接口	描述
adjustAudioFilePublishVolume	调整远端音频播放的音量
getAudioFileVolume	获取音频播放的音量
getAudioFilePlayoutVolume	获取音频本地播放的音量
getAudioFilePublishVolume	获取音频远端播放的音量
getAudioFileDuration	获取音频文件的时长
getAudioFilePosition	获取音频文件当前播放位置
setAudioFilePosition	设置音频文件播放位置
playAudioClip	开始播放音效文件
stopAudioClip	停止播放音效文件
pauseAudioClip	暂停播放音效文件
resumeAudioClip	恢复播放音效文件
stopAllAudioClips	停止播放所有音效文件
pauseAllAudioClips	暂停播放所有音效文件
resumeAllAudioClips	恢复播放所有音效文件
setAudioClipsVolume	设置音效播放的最大音量
getAudioClipsVolume	获取音效播放的最大音量
setVolumeOfAudioClip	设置指定音效的播放音量
getVolumeOfAudioClip	获取指定音效的播放音量
setAudioClipPosition	设置指定音效文件的播放位置
getAudioClipCurrentPosition	获取指定音效文件当前的播放位置
getAudioClipDuration	获取音效的文件时长
preloadAudioClip	预加载音效文件
unloadAudioClip	删除预加载音效文件

音频增强管理

表 5-8 音频增强接口

接口	描述
enableUserVolumeNotify	设置音量值上报回调函数 onUserVolumeStatsNotify回调周期

检测功能

表 5-9 检测功能接口

接口	描述
startNetworkTest	开启会前网络探测
stopNetworkTest	停止会前网络检测

自定义视频采集和渲染

表 5-10 自定义视频采集和渲染接口

接口	描述
setExternalVideoFrameOutputEnable	设置视频数据输出使能
setExternalDataFrameOutputEnable	设置共享数据输出使能
setExternalVideoCapture	设置是否开启外部视频采集
pushExternalVideoFrame	输入外部视频数据
pushAuxExternalVideoFrame	辅流输入外部视频数据

自定义音频采集和渲染

表 5-11 自定义音频采集和渲染接口

接口	描述
setExternalAudioFrameOutputEnable	设置音频数据输出使能
setExternalAudioCapture	设置是否开启外部音频采集
pushExternalAudioFrame	输入外部音频数据

设备管理

表 5-12 设备管理接口

接口	描述
setCameraConfig	设置摄像头参数
switchCamera	切换摄像头

接口	描述
setSpeakerModel	设置声音播放模式
setDefaultSpeakerModel	设置默认的声音播放模式
setLayoutDirect	设置显示模式，区分横屏还是竖屏，用于保证摄像头方向与本地界面方向一致
isSpeakerphoneEnabled	查询是否启用扬声器

5.4.1.2 初始化等基础接口

create

```
public static synchronized HRTCEngine create(HRTCEngineConfig config, IHRTCEngineEventHandler eventHandler)
```

【功能说明】

创建SparkRTC引擎实例。

【请求参数】

- config：引擎创建相关参数，具体请参见HRTCEngineConfig。
- eventHandler：引擎事件句柄，用于加入房间、离开房间等事件回调，具体请参见IHRTCEngineEventHandler。

【返回参数】

返回引擎实例对象。

destroy

```
public static synchronized void destroy()
```

【功能说明】

销毁SparkRTC引擎。

【请求参数】

无

【返回参数】

无

⚠ 注意

请不要在RTC SDK的接口回调函数中直接调用此方法，请切回业务线程后调用。

logUpload

```
public abstract int logUpload();
```

【功能说明】

上传日志。

【请求参数】

无

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

⚠ 注意

- 加入房间成功后才能主动上传日志。
- 会触发以下回调：
  - **onLogUploadResult**：日志上传结果回调。
  - **onLogUploadProgress**：日志上传进度回调。

disableRejoinRoom

```
public abstract int disableRejoinRoom(boolean disable);
```

【功能说明】

设置是否禁用房间重连功能。

【请求参数】

disable：true表示禁用，false表示不禁用。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

enableStats

```
public abstract int enableStats(boolean enabled);
```

【功能说明】

打点功能开关，在**create**之后调用。默认开启。

【请求参数】

enabled：是否打开打点功能开关，true表示打开，false表示关闭。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

getVersion

```
public static String getVersion();
```

【功能说明】

获取SDK版本号。

【返回参数】

SDK版本号。

setEncryption

```
public abstract int setEncryption(HRTCEncryptionConfig encryptionParam);
```

【功能说明】

设置端到端加密方式。需要在加入房间前设置生效。其中sdk加密模式，需要设置16位加密密钥和加密算法，app加密模式需要先设置回调接口。

【请求参数】

encryptionParam：加密配置，具体请参见**HRTCEncryptionConfig**。

【返回参数】

- 0：成功。
- <0：失败。具体请参见**客户端错误码**。

setAccessResourceType

```
public abstract int setAccessResourceType(int resType);
```

【功能说明】

设置接入的环境，不支持跨房间场景。

【请求参数】

- resType：环境类型。
- 0：公网sfu资源。
- 1：公司局Sfu。
- 2：MPC。
- 3：LLL

【返回参数】

- 0：成功。
- <0：失败。具体请参见**客户端错误码**。

setNetworkBandwidth

```
public abstract int setNetworkBandwidth(HRTCNetworkBandwidth bandwidthParam);
```

【功能说明】

设置网络带宽限制。需要在每次加入房间之前设置。

【请求参数】

bandwidthParam：带宽设置参数，具体请参见**HRTCNetWorkBandwidth**。

【返回参数】

- 0：成功。
- <0：失败。具体请参见**客户端错误码**。

5.4.1.3 房间功能

joinRoom

```
public abstract int joinRoom(HRTCJoinParam joinParam);
```

【功能说明】

加入房间。

【请求参数】

joinParam：入会参数，具体请参见**HRTCJoinParam**。

【返回参数】

- 0：成功。
- 1：失败。具体请参见**客户端错误码**。
- 2：上下文为空。

⚠ 注意

该方法将会触发以下回调：

- **onConnectionStateChangedNotify**：连接状态发送改变。
- **onJoinRoomSuccess**：加入房间成功时回调。
- **onJoinRoomFailure**：加入房间失败时回调，失败原因请参见**•加入房间失败时，如何解决?** 。
- **onRemoteUserOnline**：加入房间成功后，通知房间内已加入用户的回调，不包括自己。

leaveRoom

```
public abstract int leaveRoom()
```

【功能说明】

离开房间。

【请求参数】

无

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

⚠ 注意

会触发以下回调：

- **onLeaveRoom**：离开房间回调。
- **onConnectionStateChangedNotify**：连接状态改变回调。

renewAuthorization

```
public abstract int renewAuthorization(String signature, long ctime);
```

【功能说明】

鉴权签名过期，收到**onAuthorizationExpired**回调后更新鉴权签名。

【请求参数】

- signature：鉴权签名字符串。
- ctime：过期时间，单位：秒。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

changeUserRole

```
public abstract int changeUserRole(HRTCRoleType role, String authorization, long ctime);
```

【功能说明】

设置本端用户在房间内的角色。

【请求参数】

- role：用户角色，具体请参见**HRTCRoleType**。
- authorization：预留参数，填null。
- ctime：预留参数，填0。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

⚠ 注意

支持joiner，player角色间相互切换。

- 切换成功将触发**onUserRoleChangedNotify**回调。
- 切换失败将触发**onError**回调，返回错误码“HRTC_ERR_CODE_USER_ROLE_CHANGE_FAIL”。

changeUserName

```
public abstract int changeUserName(String usrName);
```

【功能说明】

修改用户昵称。

【请求参数】

usrName：用户新的昵称。昵称不为空，且最大不超过256。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

⚠ 注意

修改成功后，本端会回调**onUserNameChangedNotify**事件，远端会回调**onRemoteUserNameChangedNotify**事件。

createConnection

```
public abstract HRTCConnection createConnection(String roomId, IHRTCConnectionEventHandler eventHandler);
```

【功能说明】

根据房间ID，创建HRTCConnection对象，为跨房做准备。

通过此接口创建一个与房间关联的HRTCConnection连接对象。

该方法支持多次调用，创建多个HRTCConnection连接对象，调用每个对象中的joinRoom方法，可以同时加入到多个房间。在每个房间中，可以分别订阅和选看房间中的用户。

具体请参见**事件回调(IHRTCConnectionEventHandler)**中相关接口和回调。

【请求参数】

roomId：房间ID。

eventHandler：引擎事件句柄，用于加入房间、离开房间等事件回调，具体请参见**事件回调(IHRTCConnectionEventHandler)**。

【返回参数】

- 不为null：成功。
- null：失败。

⚠ 注意

- 同一时间最多只能创建4个连接对象，每个连接对象对应的房间ID必须互不相同。
- 如果使用HRTCConnection对象加入房间，则加入房间的房间ID不能和已创建连接对象对应的房间ID相同。
- 同一时间只能以JOINER角色加入某一个房间。

addMultiRoomMediaRelay

```
virtual int addMultiRoomMediaRelay(HRTCMultiRoomMediaRelayConfiguration
roomMediaRelayConfiguration)
```

【功能说明】

添加单个跨房。发起跨房后由云侧单向将本端上行流推至目标房间，即只推流不收流。

【请求参数】

roomMediaRelayConfiguration：跨房信息，具体请参见**HRTCMultiRoomMediaRelayConfiguration**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

removeMultiRoomMediaRelay

```
virtual int removeMultiRoomMediaRelay(HRTCMultiRoomMediaRelayConfiguration
roomMediaRelayConfiguration)
```

【功能说明】

删除单个跨房。

【请求参数】

roomMediaRelayConfiguration：跨房信息，具体请参见**HRTCMultiRoomMediaRelayConfiguration**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

stopMultiRoomMediaRelay

```
virtual int stopMultiRoomMediaRelay()
```

【功能说明】

停止所有跨房。

【请求参数】

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

5.4.1.4 音频管理

muteLocalAudio

```
public abstract int muteLocalAudio(boolean mute);
```

【功能说明】

设置是否发送本地音频流。

【请求参数】

mute：true表示不发送，false表示发送。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

注意

成功加入房间发送音频流后才能调用，默认为发送本地音频流。

muteRemoteAudio

```
public abstract int muteRemoteAudio(String userId, boolean mute);
```

【功能说明】

设置是否接收对应远端用户的音频流。

【请求参数】

- userId：用户ID。
- mute：true表示取消音频流接收，false表示开启音频流接收，默认为false。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

muteAllRemoteAudio

```
public abstract int muteAllRemoteAudio(boolean mute);
```

【功能说明】

设置是否接收所有远端用户的音频流。

【请求参数】

mute：true表示取消音频流接收，false表示开启音频流接收。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

注意

- 取消所有音频流接收，同时也会取消接收新加入用户的音频流。
- 开启所有音频流接收，同时也会开启接收新加入用户的音频流。
- 默认开启所有音频流接收。

enableLocalAudioStream

```
public abstract int enableLocalAudioStream(boolean enabled);
```

【功能说明】

设置是否开启音频采集。

【请求参数】

enabled：true表示采集开启，false表示关闭。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

⚠ 注意

默认开启，本端调用该接口时，远端用户会触发**onRemoteAudioStateChangedNotify**远端音频流状态变化回调。

adjustRecordingVolume

```
public abstract int adjustRecordingVolume(int volume);
```

【功能说明】

调整录制音量值。

【请求参数】

volume：音量值，取值范围：[0,100]，默认音量值为10， 此接口不会影响系统音量。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

adjustPlaybackVolume

```
public abstract int adjustPlaybackVolume(int volume);
```

【功能说明】

调整播放音量值。

【请求参数】

volume：音量值，取值范围为[0,100]，默认音量值为10，此接口不会影响系统音量。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

adjustPlaybackVolume

```
public abstract int adjustPlaybackVolume(String userId, int volume);
```

【功能说明】

调整单个用户播放音量增益值。

【请求参数】

- userId：用户ID。
- volume：音量值，取值范围为[0,100]，默认音量值为10无增益，10以下表示负增益，10以上表示正增益，此接口不会影响系统音量。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

sendAudioSeiMsg

```
public abstract int sendAudioSeiMsg(String message, int repeateCount);
```

【功能说明】

发送音频SEI消息。通过音频SEI可将自定义信息嵌入到音频流中，发送给其他用户。

【请求参数】

- message：发送的内容。长度为1-500字节。
- repeateCount：发送次数（1-10），根据需要填发送次数，一般发1次。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

setAudioFrameRecordParameters

```
public abstract int setAudioFrameRecordParameters(int sampleRate, int channel, HRTCAudioOperateMode mode, int samplesPerCall);
```

【功能说明】

设置采集回调参数，配合**setAudioFrameObserver**的onAudioFrameRecord使用。

【请求参数】

- sampleRate：**onAudioFrameRecord**中返回的采样率，可设置为8000，16000，32000, 44100，48000。
- channel：声道，1表示单声道，2表示双声道。
- mode：可读可写模式，具体请参见**HRTCAudioOperateMode**。
- samplesPerCall：每次回调的单声道样点数（小于(sampleRate/100)*channel*2*3，大于(sample/(100*3))*channel*2）。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

setAudioConfig

```
public abstract int setAudioConfig(HRTCAudioQualityLevel level, HRTCAudioSceneType scene)
```

【功能说明】

设置音频使用场景。该接口需要在**joinRoom**前调用。

此接口可在create接口设置场景后改变音频场景，暂不支持初始化scene设置音乐再通过此接口设置为会议。

【请求参数】

- level：表示档位，会议模式暂时只支持16k。
- scene：表示音频场景模式。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

5.4.1.5 视频管理

createRenderer

```
public abstract SurfaceView createRenderer(Context context)
```

【功能说明】

创建渲染视图。

【请求参数】

context：上下文。

【返回参数】

创建的渲染视图。

updateLocalRenderMode

```
public abstract int updateLocalRenderMode(HRTCVideoDisplayMode displayMode, HRTCVideoMirrorType mirrorMode) = 0;
```

【功能说明】

设置本地窗口显示模式，镜像模式。

【请求参数】

- displayMode：显示模式，具体请参见**HRTCVideoDisplayMode**。
- mirrorMode：镜像模式，具体请参见**HRTCVideoMirrorType**。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

setupLocalView

```
public abstract int setupLocalView(SurfaceView view);
public abstract int setupLocalView(SurfaceView view, HRTCVideoDisplayMode viewMode);
```

【功能说明】

设置本地窗口。

【请求参数】

- view：窗口视图。
- displayMode：显示模式，具体请参见**HRTCVideoDisplayMode**。不设置则默认为裁剪模式。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

pushLocalVideo

```
public abstract int pushLocalVideo(boolean push);
```

【功能说明】

设置是否发送本地视频流。

【请求参数】

push：true表示发送，false表示不发送。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

setVideoEncoderConfig

```
public abstract int setVideoEncoderConfig(int totalBitRate, List<HRTCVideoEncParam> encoderParams);
public abstract int setVideoEncoderConfig(HRTCVideoEncParam encoderParam);
```

【功能说明】

设置视频编码参数。具体请参见全平台**setVideoEncoderConfig**。

【请求参数】

- totalBitRate：视频最大码率，默认值4096。
- encoderParams：视频编码参数列表，具体请参见**HRTCVideoEncParam**。
- encoderParam：视频编码参数，具体请参见**HRTCVideoEncParam**。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

⚠ 注意

- 安卓移动端采集帧率限制20帧，建议设置的编码帧率不要超过20帧

setNonStandardVideoEncoder

```
public abstract int setNonStandardVideoEncoder(List<HRTCVideoEncParam> encParams);
```

【功能说明】

设置非标视频编码参数。参数列表，第一个为大流（必填），第二个为小流（选填）。

【请求参数】

encoderParams：视频编码参数列表，具体请参见**HRTCVideoEncParam**。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

setRemoteVideoAdjustResolution

```
public abstract int setRemoteVideoAdjustResolution(boolean enable);
```

【功能说明】

设置是否开启远端流分辨率自适应。默认开启自适应。

【请求参数】

enable：是否开启分辨率自适应。默认开启。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

startLocalPreview

```
public abstract int startLocalPreview();
```

【功能说明】

开始本地预览。

【请求参数】

无

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

⚠ 注意

- 该接口限制在房间外调用，在房间内设置不生效，需要调用stopLocalPreview关闭预览，否则将一直处于预览状态。
- 在房间内预览：可调用**setupLocalView**设置有效view开启预览，设置为null表示关闭预览，不需要调用stopLocalPreview。
- 在房间外预览：先调用**setupLocalView**设置有效view，再调用startLocalPreview开启预览；关闭时，先调用**setupLocalView**设置为null，再调用stopLocalPreview关闭预览。

stopLocalPreview

```
public abstract int stopLocalPreview();
```

【功能说明】

停止本地预览。

【请求参数】

无

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

⚠ 注意

如果调用了**startLocalPreview**，需调用stopLocalPreview关闭预览，否则将一直处于预览状态。该接口限制在房间外调用，在房间内设置不生效。

startRemoteStreamView

```
public abstract int startRemoteStreamView(String userId, SurfaceView view, HRTCStreamType streamType,
boolean disableAdjustRes);
```

【功能说明】

开始订阅远端视频流，并设置远端窗口。

【请求参数】

- userId：用户ID。
- view：远端窗口视图。
- streamType：流类型，具体请参见**HRTCStreamType**。
- disableAdjustRes：禁用分辨率自适应的标志。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

stopRemoteStreamView

```
public abstract int stopRemoteStreamView(String userId);
```

【功能说明】

停止订阅远端视频流，并关闭远端窗口。

【请求参数】

userId：用户ID。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

setupRemoteView

```
public abstract int setupRemoteView(String userId, SurfaceView view);
```

【功能说明】

设置远端流视图，该接口不影响收流。

【请求参数】

- userId：远端用户的唯一标识。
- view：远端窗口视图，view为null时，解除窗口绑定。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

updateRemoteRenderMode

```
public abstract int updateRemoteRenderMode(String userId, HRTCVideoDisplayMode displayMode,
HRTCVideoMirrorType mirrorMode);
```

【功能说明】

设置远端窗口渲染模式。

【请求参数】

- userId：用户ID。
- displayMode：渲染模式，具体请参见**HRTCVideoDisplayMode**。
- mirrorMode：镜像模式，具体请参见**HRTCVideoMirrorType** 。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

pullRemoteVideo

```
public abstract int pullRemoteVideo(String userId, boolean pull);
```

【功能说明】

开启、关闭指定远端用户的视频流。

【请求参数】

- userId：远端用户的userId，唯一标识。
- pull：true表示开始接收，false表示关闭接收。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

pullAllRemoteVideo

```
public abstract int pullAllRemoteVideo(boolean pull);
```

【功能说明】

批量开启、关闭当前所有远端用户的视频流。

【请求参数】

pull：true表示开启接收，false表示关闭接收， 默认开启接收。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

⚠ 注意

开启本地镜像后，本地视频窗口看到的是镜像视图，但不影响远端查看自己的视图。

setVideoEncoderMirror

```
public abstract int setVideoEncoderMirror(HRTCVideoMirrorType mirrorType);
```

【功能说明】

设置视频编码镜像模式。

【请求参数】

mirrorType：镜像模式，是否开启镜像，具体请参见**HRTCVideoMirrorType**，默认值false。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

⚠ 注意

开启镜像后，仅改变编码发送给远端用户的视图，不影响自己看到的本地视频窗口。

enableLocalVideo

```
public abstract int enableLocalVideo(boolean enabled);
```

【功能说明】

设置是否开启摄像头采集视频。

【请求参数】

enabled：true表示开启，false表示关闭。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

⚠ 注意

默认开启，本端调用时，远端用户会触发**onRemoteVideoStateChangedNotify**远端视频流状态变化回调。

---

enableVideoSuperResolution

```
public abstract int enableVideoSuperResolution(boolean enabled);
```

【功能说明】

设置是否开启视频超分 。

【请求参数】

enabled：true表示开启，false表示关闭。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

enableSmallVideoStream

```
public abstract int enableSmallVideoStream(boolean enable, HRTCVideoEncParam encoderParam);
```

【功能说明】

大小流模式设置是否开启小流，并设置编码参数。小流选择性开启。

【请求参数】

- enable：是否开启小流。
- encoderParam：视频编码参数。包括流类型、宽、高、码率、帧率等。具体请参见**HRTCVideoEncParam**。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

setPriorRemoteVideoStreamType

```
public abstract int setPriorRemoteVideoStreamType(HRTCVideoStreamType type);
```

【功能说明】

大小流模式，设置所有订阅的远端视频流类型。默认订阅大流，优先应用setRemoteVideoStreamType接口设置的用户流类型。

【请求参数】

type：订阅的视频流类型，分为大流和小流，具体请参见**HRTCVideoStreamType**。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

setRemoteVideoStreamType

```
public abstract int setRemoteVideoStreamType(String userId, HRTCVideoStreamType type);
```

【功能说明】

大小流模式，设置指定订阅的远端视频流类型。

【请求参数】

- userId：远端用户唯一标识。
- type：订阅的视频流类型，分为大流和小流，具体请参见**HRTCVideoStreamType**。

startPublishStream

```
public abstract int startPublishStream(String taskId, List<String> urlList, HRTCTranscodeConfig
transcodeConfig);
```

【功能说明】

开始旁路推流。

【请求参数】

- taskId：任务id，业务自行定义，保证唯一。
- urlList：url数组。参考**HRTCRtmpUrlList**
- transcodeConfig：用户id数组和其他参数，具体请参见**HRTCTranscodeConfig**。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

updateTransCoding

```
public abstract int updateTransCoding(String taskId, HRTCTranscodeConfig transcodeConfig);
```

【功能说明】

更新旁路推流。

【请求参数】

- taskId：任务id，业务自行定义，保证唯一。
- transcodeConfig：用户id数组和其他参数，具体请参见**HRTCTranscodeConfig**。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

stopPublishStream

```
virtual int stopPublishStream(const char* taskId)
```

【功能说明】

停止旁路推流。

【请求参数】

taskId：任务id，支持自定义，需保证唯一性。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

startAllRemoteView

```
public abstract int startAllRemoteView(int counts, List<HRTCVideoRemoteView> viewInfoList);
```

【功能说明】

批量设置远端流视图。

【请求参数】

- counts：必选，number类型，为数组的长度；如果设置为0，则取消所有远端流视图，大于0，则取消没选中用户的远端视图。
- viewInfoList：订阅的视图信息，主要包括该视图的句柄、流类型、用户ID、是否自适应等，具体请参见**HRTCVideoRemoteView**。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

5.4.1.6 屏幕共享

startScreenShare

```
public abstract int startScreenShare();
```

【功能说明】

开启屏幕共享。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

⚠ 注意

- Android 5.0及以上版本支持此功能。
- 使用时，在AndroidManifest.xml中需添加如下约束。
```
<activity
<activity android:name="com.huawei.allplatform.screencapture.HRTCScreenShareAssistantActivity"
    android:theme="@style/dialog" />
```
- targetSdkVersion >= 24时，在AndroidManifest.xml中需添加如下约束，并在开启屏幕共享时弹出自定义悬浮窗，避免被系统强杀掉。
```
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
```
- targetSdkVersion >= 29时，在AndroidManifest.xml中需添加如下约束。
```
<service android:name="com.huawei.allplatform.screencapture.HRTCScreenShareService"
    android:enabled="true"
    android:foregroundServiceType="mediaProjection"/>
```
- 目前可支持多路辅流共享，若需开启多辅流，请**提交工单**联系技术支持处理。
- 开启后将触发**onScreenShareStarted**回调。
- 远端会收到**onUserAuxiliaryStreamAvailable**通知，可据此发起辅流选看。

stopScreenShare

```
public abstract int stopScreenShare();
```

【功能说明】

停止屏幕共享。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

⚠ 注意

- SDK 1.7.1及以上版本支持。
- 停止后将会触发**onScreenShareStopped**回调。

5.4.1.7 辅流管理

setAuxiliaryVideoEncodeSmooth

```
public abstract int setAuxiliaryVideoEncodeSmooth(boolean enabled);
```

【功能说明】

设置是否开启辅流的流畅度优先。

【请求参数】

enable：true表示辅流分辨率为720p，false表示辅流分辨率为1080p。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

⚠ 注意

- SDK 1.7.1及以上版本支持。
- 开启后，辅流发流分辨率为720p，否则发流分辨率为1080p。默认不开启。

startRemoteAuxiliaryStreamView

```
public abstract int startRemoteAuxiliaryStreamView(String userId, SurfaceView view);
```

【功能说明】

开始订阅辅流。

【请求参数】

- userId：用户ID。
- view：窗口视图。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

⚠ 注意

- 收到**onUserAuxiliaryStreamAvailable**通知后，获取对应的userId。
- 多辅流场景下，一个用户只能同时订阅一条辅流。即当前正在订阅用户A的辅流，需要订阅另一个用户B的辅流时，需要调用startRemoteAuxiliaryStreamView停止订阅用户A的辅流后，才能订阅用户B的辅流。

stopRemoteAuxiliaryStreamView

```
public abstract int stopRemoteAuxiliaryStreamView(String userId);
```

【功能说明】

停止订阅辅流。

【请求参数】

userId：用户ID。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

setRemoteAuxiliaryStreamViewRotation

```
public abstract int setRemoteAuxiliaryStreamViewRotation(String userId, HRTCRotationType rotation);
```

【功能说明】

设置辅流角度。

【请求参数】

- userId：用户ID。
- rotation：辅流角度，默认值为HRTC_ROTATION_TYPE_0，具体请参见**HRTCRotationType**。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

updateRemoteAuxiliaryStreamRenderMode

```
public abstract int updateRemoteAuxiliaryStreamRenderMode(String userId, HRTCVideoDisplayMode displayMode, HRTCVideoMirrorType mirrorMode);
```

【功能说明】

设置辅流渲染模式。

【请求参数】

- userId：用户ID。
- displayMode：渲染模式，默认值为HRTC_VIDEO_DISPLAY_MODE_FIT，具体请参见**HRTCVideoDisplayMode**。
- mirrorMode：镜像模式，默认值为HRTC_VIDEO_MIRROR_TYPE_DISABLE，具体请参见**HRTCVideoMirrorType**。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

setAuxiliaryVideoEncoderConfig

```
public abstract int setAuxiliaryVideoEncoderConfig(HRTCVideoAuxiliaryEncParam encoderParams);
```

【功能说明】

设置辅流编码参数。

【请求参数】

encoderParams：需要设置的辅流编码参数，包括宽、高、帧率和码率，具体请参见**HRTCVideoAuxiliaryEncParam**。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

⚠ 注意

使用该接口设置辅流编码参数时，由于当前系统策略会根据获取的区域宽高比对设置的宽高进行调整，使用用户设置的宽高比与获取的宽高比保持一致，因此用户实际收流的分辨率与设置的分辨率可能存在不同。

5.4.1.8 音效文件播放管理

startAudioFile

```
public abstract int startAudioFile(String fullFilePath, int publish, int cycle, int replace);
public abstract int startAudioFile(String fullFilePath, int publish, int cycle, int replace, int startPos);
```

【功能说明】

开始播放音频文件。当前支持本端播放和在线播放。支持的格式包括：mp3、flac、mp4、m4a、aac、3gp、mkv、wav、amr、pcm、ogg。

【请求参数】

- fullFilePath：音频文件的本地全路径，支持播放本地文件或网络文件。
- publish：播放模式，0表示只有本端能听到播放的音频，1表示远端也能听到播放的音频。
- cycle：循环次数，0表示无限循环。
- replace：是否替代麦克风采集，等于1时用音频文件的声音替换麦克风采集的声音。
- startPos：音频文件开始播放的位置，单位为ms。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

⚠ 注意

会触发**onAudioMixStateChangedNotify**回调。

stopAudioFile

```
public abstract int stopAudioFile();
```

【功能说明】

停止播放音频文件。

【请求参数】

无

【返回参数】

- 0：成功。
- > 0：失败。

注意

会触发**onAudioMixStateChangedNotify**回调。

pauseAudioFile

```
public abstract int pauseAudioFile();
```

【功能说明】

暂停播放音频文件。

【请求参数】

无

【返回参数】

- 0：成功。
- >0：失败。

注意

会触发**onAudioMixStateChangedNotify**回调。

resumeAudioFile

```
public abstract int resumeAudioFile();
```

【功能说明】

恢复播放音频文件。

【请求参数】

无

【返回参数】

- 0：成功。
- >0：失败。

注意

会触发**onAudioMixStateChangedNotify**回调。

isPlayMixMyself

```
public abstract int isPlayMixMyself(boolean myself);
```

【功能说明】

是否只有本地可以听到混音。

【请求参数】

myself：true表示只有本地可以听到混音， false表示本地和对方都可以听到混音，默认值为false。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

isMixWithMicrophone

```
public abstract int isMixWithMicrophone(boolean withMic);
```

【功能说明】

是否需要替代采集。

【请求参数】

withMic：true表示替代采集，远端只能听到音乐，false表示不替代，远端可以同时听到音乐和MIC采集的声音，默认值为false。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

adjustAudioFileVolume

```
public abstract int adjustAudioFileVolume(int volume);
```

【功能说明】

调整本地和远端音频播放的音量。

【请求参数】

volume：音量大小，范围为0-100，默认值为100。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**客户端错误码**。

adjustAudioFilePlayoutVolume

```
public abstract int adjustAudioFilePlayoutVolume(int volume);
```

【功能说明】

调整本地音频播放的音量。

【请求参数】

volume：音量大小，范围为0-100，默认值为100。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**客户端错误码**。

adjustAudioFilePublishVolume

```
public abstract int adjustAudioFilePublishVolume(int volume);
```

【功能说明】

调整远端音频播放的音量。

【请求参数】

volume：音量大小，范围0-100，默认值100。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**客户端错误码**。

getAudioFileVolume

```
public abstract int getAudioFileVolume();
```

【功能说明】

获取音频播放的音量。

【请求参数】

无

【返回参数】

- >=0：音量大小，范围为0-100。
- < 0：方法调用失败。具体请参见**客户端错误码**。

getAudioFilePlayoutVolume

```
public abstract int getAudioFilePlayoutVolume();
```

【功能说明】

获取音频本地播放的音量。

【请求参数】

无

【返回参数】

- >=0：音量大小，范围为0-100。
- < 0：方法调用失败。具体请参见**客户端错误码**。

getAudioFilePublishVolume

```
public abstract int getAudioFilePublishVolume();
```

【功能说明】

获取音频远端播放的音量。

【请求参数】

无

【返回参数】

- >=0：音量大小，范围为0-100。
- < 0：方法调用失败。具体请参见**客户端错误码**。

getAudioFileDuration

```
public abstract int getAudioFileDuration();
```

【功能说明】

获取音频文件的时长。

【请求参数】

无

【返回参数】

- >0：音频时长，单位为ms。
- <= 0：方法调用失败。具体请参见**客户端错误码**。

getAudioFilePosition

```
public abstract int getAudioFilePosition();
```

【功能说明】

获取音频文件当前播放位置。

【请求参数】

无

【返回参数】

- >=0：播放位置，单位为ms。
- < 0：方法调用失败。具体请参见**客户端错误码**。

setAudioFilePosition

```
public abstract int setAudioFilePosition(int position);
```

【功能说明】

设置音频文件播放位置。

【请求参数】

position：播放位置，单位为ms。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**客户端错误码**。

playAudioClip

```
public abstract int playAudioClip(int soundId, String filePath, int loop, double pitch, double pan, double gain, int publish, int startPos);
```

【功能说明】

播放音效文件并启动混音，需要在有joiner加入房间后调用。

【请求参数】

- soundId：音效ID，取值>= 0。
- filePath：音效文件路径，支持本地文件和网络文件。
- loop：音效文件播放次数，0为循环播放。
- pitch：音调大小，当前不支持。
- pan：空间位置，当前不支持。
- gain：音量大小，取值范围0-100。
- publish：1表示将音效文件混音后发送到远端，0为本地播放，不发送到远端。
- startPos：起始播放位置，单位为ms。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**客户端错误码**。

⚠ 注意

soundId需要开发者自己生成并维护，保证不同的soundId对应不同的音效播放实例。同时音效播放完毕或者停止播放后，soundId最好主动回收，下一次播放音效的时候，尽量复用被回收的soundId。

stopAudioClip

```
public abstract int stopAudioClip(int soundId);
```

【功能说明】

停止播放指定的音效文件。

【请求参数】

soundId：音效ID，取值>= 0。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**客户端错误码**。

pauseAudioClip

```
public abstract int pauseAudioClip(int soundId);
```

【功能说明】

暂停播放指定的音效文件。

【请求参数】

soundId：音效ID，取值>= 0。

【返回参数】

- 0：方法调用成功。

- < 0：方法调用失败。具体请参见**客户端错误码**。

resumeAudioClip

```
public abstract int resumeAudioClip(int soundId);
```

【功能说明】

恢复播放指定的音效文件。

【请求参数】

soundId：音效ID，取值>= 0。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**客户端错误码**。

stopAllAudioClips

```
public abstract int stopAllAudioClips();
```

【功能说明】

停止播放所有音效文件。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**客户端错误码**。

pauseAllAudioClips

```
public abstract int pauseAllAudioClips();
```

【功能说明】

暂停播放所有音效文件。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**客户端错误码**。

resumeAllAudioClips

```
public abstract int resumeAllAudioClips();
```

【功能说明】

恢复播放所有音效文件。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**客户端错误码**。

setAudioClipsVolume

```
public abstract int setAudioClipsVolume(double volume);
```

【功能说明】

设置音效播放的最大音量。

【请求参数】

volume：音量大小，范围为0-100。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**客户端错误码**。

getAudioClipsVolume

```
public abstract int getAudioClipsVolume();
```

【功能说明】

获取音效播放的最大音量。

【请求参数】

无

【返回参数】

- >= 0：音量大小，范围为0-100。
- < 0：方法调用失败。具体请参见**客户端错误码**。

⚠ 注意

音效文件支持同时播放多个，**setAudioClipsVolume**接口设置的音量是所有音频文件的最大音量，**setVolumeOfAudioClip**接口设置的是单个音效文件的音量，音效文件的实际播放音量 = 最大音量 * 自身音量 / 100。例如，最大音量是50，单个音效音量是80，实际播放音量就是50 * 80 / 100 = 40。

setVolumeOfAudioClip

```
public abstract int setVolumeOfAudioClip(int soundId, double volume);
```

【功能说明】

设置指定音效的播放音量。

【请求参数】

- soundId：音效ID，取值>= 0。
- volume：音量大小，范围为0-100，默认值为100。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**客户端错误码**。

getVolumeOfAudioClip

```
public abstract int getVolumeOfAudioClip(int soundId);
```

【功能说明】

获取指定音效的播放音量。

【请求参数】

soundId：音效ID，取值>= 0。

【返回参数】

- >= 0：音量大小，范围为0-100，默认值为100。
- < 0：方法调用失败。具体请参见**客户端错误码**。

⚠ 注意

支持同时播放多个音效文件，**setAudioClipsVolume**接口设置的音量是所有音频文件的最大音量，**setVolumeOfAudioClip**接口设置的是单个音效文件的音量，音效文件的实际播放音量 = 最大音量 * 自身音量 / 100。例如，最大音量是50，单个音效音量是80，实际播放音量就是50 * 80 / 100 = 40。

setAudioClipPosition

```
public abstract int setAudioClipPosition(int soundId, int pos);
```

【功能说明】

设置指定音效的播放位置。

【请求参数】

- soundId：音效ID，取值>= 0。
- pos：播放位置，单位为ms。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**客户端错误码**。

getAudioClipCurrentPosition

```
public abstract int getAudioClipCurrentPosition(int soundId);
```

【功能说明】

获取指定音效文件当前的播放位置。

【请求参数】

soundId：音效ID，取值>= 0。

【返回参数】

- >= 0：播放位置，单位为ms。
- < 0：方法调用失败。具体请参见**客户端错误码**。

getAudioClipDuration

```
public abstract int getAudioClipDuration(String filePath);
```

【功能说明】

获取音效的文件时长。

【请求参数】

filePath：音效文件路径，支持本地文件和网络文件。

【返回参数】

- > 0：音效文件时长，单位为ms。
- <= 0：方法调用失败。具体请参见**客户端错误码**。

preloadAudioClip

```
public abstract int preloadAudioClip(int soundId, String filePath);
```

【功能说明】

预加载音效文件，当前不支持。

【请求参数】

- soundId：音效ID，取值>= 0。
- filePath：音效文件路径。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**客户端错误码**。

unloadAudioClip

```
public abstract int unloadAudioClip(int soundId);
```

【功能说明】

删除预加载的音效文件。

【请求参数】

soundId：音效ID，取值>= 0。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**客户端错误码**。

5.4.1.9 音频增强管理

enableUserVolumeNotify

```
public abstract int enableUserVolumeNotify(int interval);
```

【功能说明】

设置音量值上报回调函数(onUserVolumeStatsNotify)的回调周期。

【请求参数】

interval：音量值上报周期，默认关闭音量回调。

- 0：关闭音量回调。
- [100，10000]：有效值范围，单位为毫秒，建议设置为2000ms，默认值为2000ms。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**客户端错误码**。

5.4.1.10 检测功能

startNetworkTest

```
public abstract int startNetworkTest(HRTCNetworkTestConfig networkTestConfig);
```

【功能说明】

启动入会前网络检测。

【请求参数】

networkTestConfig：检测配置信息，具体请参见**HRTCNetworkTestConfig**。

【返回参数】

- 0：表示调用启动命令成功。
- > 0：表示调用启动命令失败。具体请参见**客户端错误码**。

stopNetworkTest

```
public abstract int stopNetworkTest();
```

【功能说明】

停止入会前网络检测。

【请求参数】

无

【返回参数】

- 0：表示调用停止命令成功。
- > 0：表示调用停止命令失败。具体请参见**客户端错误码**。

5.4.1.11 自定义音频采集和渲染

setExternalAudioFrameOutputEnable

```
public abstract int setExternalAudioFrameOutputEnable(boolean localEnable,boolean remoteEnable);
```

【功能说明】

设置音频数据输出使能。

【请求参数】

- localEnable：true表示输出本地音频数据，false表示不输出本地音频数据。
- remoteEnable：true表示输出远端音频数据，false表示不输出远端音频数据。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

注意

通过**onPlaybackExternalAudioFrame**回调音频数据。

setExternalAudioCapture

```
public abstract int setExternalAudioCapture(boolean audioEnable,int sampleRate,int channels);
```

【功能说明】

设置是否开启外部音频采集。需要在加入房间前调用。

【请求参数】

- audioEnable：true表示音频使用外部采集，false表示音频不使用外部采集，默认值为false。
- sampleRate：采样率，当前支持16k、24k、32k、44.1k、48k采样率。
- channels：频道数，当前只支持单声道，1表示单声道，2表示双声道。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

注意

如果使用外部输入音频数据，需要在加入房间后，按照一定时间间隔调用**pushExternalAudioFrame**接口输入音频数据。

pushExternalAudioFrame

```
public abstract int pushExternalAudioFrame(byte[] audioData);
```

【功能说明】

输入外部音频数据。

【请求参数】

audioData：音频数据。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

⚠ 注意

- 此方法调用前，需要先调用**setExternalAudioCapture**设置开启外部音频采集。
- 数据输入周期：10ms。
- 音频输入数据大小：10 * sampleRate * channels * 16 / 8 / 1000。

5.4.1.12 自定义视频采集和渲染

setExternalVideoFrameOutputEnable

```
public abstract int setExternalVideoFrameOutputEnable(boolean localEnable,boolean remoteEnable);
public abstract int setExternalVideoFrameOutputEnable(boolean localEnable, boolean remoteEnable,
HRTCImageBufferFormat format);
```

【功能说明】

设置视频数据输出使能。

【请求参数】

- localEnable：true表示输出本地视频数据，false表示不输出本地视频数据，默认值为false。
- remoteEnable：true表示输出远端视频数据，false表示不输出远端视频数据，默认值为false。
- format：自渲染输出的视频帧图片格式，在**onRenderExternalVideoFrame**接口参数的videoFrameType中体现，取值请参见**HRTCImageBufferFormat**。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

setAuxiliaryExternalVideoFrameOutputEnable

```
public abstract int setAuxiliaryExternalVideoFrameOutputEnable(boolean localEnable,boolean
remoteEnable);
```

【功能说明】

设置共享辅流数据输出使能。Android平台接口暂不提供本地共享数据的设置。

【请求参数】

- localEnable：true表示输出本地共享数据，false表示不输出本地共享数据，默认值为false。

- remoteEnable：true表示输出远端共享数据，false表示不输出远端共享数据，默认值为false。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

setExternalVideoCapture

```
public abstract int setExternalVideoCapture(boolean videoEnable, HRTCVideoFrameFormat format);
```

【功能说明】

设置是否开启视频外部采集。需要在加入房间前调用。

【请求参数】

- videoEnable：true表示视频使用外部采集，false表示视频不使用外部采集，默认值为false。
- format：设置外部采集的视频格式，默认为I420（即yuv420P）。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

⚠ 注意

如果使用外部输入视频数据，需要在加入房间后，按照一定时间间隔调用**pushExternalVideoFrame**接口输入视频数据。

format可选格式为**yuv420p**、**rgba**和**texture2d**，如果需要外部传入texture2d编码的数据，需要设置format为texture2d，否则传入yuv420p或者rgba。texture2d视频数据，kirin系列970以上芯片走texture2d硬编，其他芯片都是走texture2d软编。

setAuxExternalVideoCapture

```
public abstract int setAuxExternalVideoCapture(boolean videoEnable, HRTCVideoFrameFormat format);
```

【功能说明】

设置是否开启视频辅流外部采集。需要在加入房间前调用。

【请求参数】

- videoEnable：true表示视频使用辅流外部采集，false表示视频不使用辅流外部采集。
- format：设置外部采集的视频格式，默认为I420（即yuv420P）。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

⚠ 注意

此接口与屏幕共享功能互斥，若videoEnable为true，则不能使用屏幕采集功能。

format支持texture2d，如果需要外部传入texture2d编码的数据，需要设置format为texture2d。

pushExternalVideoFrame

```
public abstract int pushExternalVideoFrame(HRTCVideoFrame videoFrame);
```

【 功能说明 】

输入外部视频数据。

【 请求参数 】

videoFrame：视频数据，具体请参见**HRTCVideoFrame**。

【 返回参数 】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

⚠ 注意

需要按照一定的时间间隔进行输入，例如，视频原本为15帧/秒，则需要间隔1000/15毫秒间隔输入视频数据。

pushAuxExternalVideoFrame

```
public abstract int pushAuxExternalVideoFrame(HRTCVideoFrame videoFrame);
```

【 功能说明 】

辅流输入外部视频数据。

【 请求参数 】

videoFrame：视频数据，具体请参见**HRTCVideoFrame**。

【 返回参数 】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

⚠ 注意

此接口与屏幕采集功能互斥。

如果需要传输texture2d编码流，需要设置**setAuxExternalVideoCapture** videoEnable参数为true，format设置为texture2d。

5.4.1.13 设备管理

isSpeakerphoneEnabled

```
int isSpeakerphoneEnabled();
```

【功能说明】

是否启用扬声器。

【请求参数】

无

【返回参数】

- 0：不使用扬声器。
- 1：使用扬声器。

注意

仅支持iOS和Android。

setCameraConfig

```
public abstract int setCameraConfig(HRTCCameraConfig config);
```

【功能说明】

设置摄像头相关参数，开启摄像头前设置有效。

【请求参数】

config：摄像头的配置参数，具体请参见**HRTCCameraConfig**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**客户端错误码**。

switchCamera

```
public abstract int switchCamera();
```

【功能说明】

切换摄像头，开启摄像头后，调用生效。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**客户端错误码**。

setSpeakerModel

```
public abstract int setSpeakerModel(HRTCSpeakerModel speakerModel);
```

【功能说明】

设置声音播放模式。成功加入房间后才能调用。

【请求参数】

speakerModel：声音播放模式，默认值为HRTC_SPEAKER_MODE_SPEAKER，具体请参见**HRTCSpeakerModel**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**客户端错误码**。

setDefaultSpeakerModel

```
public abstract int setDefaultSpeakerModel(HRTCSpeakerModel speakerModel);
```

【功能说明】

设置默认的声音播放模式，在房间外调用。

【请求参数】

speakerModel：声音播放模式，具体请参见**HRTCSpeakerModel**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**客户端错误码**。

setLayoutDirect

```
public abstract void setLayoutDirect(HRTCOrientationMode layoutDirect);
```

【功能说明】

设置显示模式，区分横屏和竖屏，用于保证摄像头方向与本地界面方向一致。

【请求参数】

layoutDirect：0表示横屏模式，1表示竖屏模式，2表示横屏重力感应，3表示重力感应，4表示竖屏模式且不会随设备旋转改变采集角度，5表示adaptive模式，适用非dns模式推流外的大部分场景，能自动适配app的布局，自动调整采集方向。

【返回参数】

无

5.4.2 事件回调（IHRTCEngineEventHandler）

本章节介绍了Android SDK的回调接口IHRTCEngineEventHandler的详情。

表 5-13 回调接口

接口	描述
onError	错误回调
onJoinRoomSuccess	加入房间成功回调
onJoinRoomFailure	加入房间失败回调
onLeaveRoom	离开房间回调
onRemoteUserOnline	用户加入房间回调
onRemoteUserOffline	用户离开房间回调
onConnectionChangedNotify	网络连接状态改变回调
onLogUploadResult	日志上传结果回调
onLogUploadProgress	日志上传进度回调
onUserRoleChangedNotify	用户角色改变回调
onAudioRouteStateChangedNotify	音频路由改变回调
onUserAuxiliaryStreamAvailable	辅流加入房间回调
onRenderExternalVideoFrame	回调远端的视频原始数据流
onPlaybackExternalAudioFrame	回调远端的音频原始数据流
onVideoStatsNotify	视频流详情，2s触发一次回调
onAudioStatsNotify	音频流详情，2s触发一次回调
onAuxiliaryStreamStatsNotify	辅流详情，2s触发一次回调
onRemoteAudioStateChangedNotify	远端音频流状态变化回调
onRemoteVideoStateChangedNotify	远端视频流状态变化回调
onRejoinRoomSuccess	重新加入房间回调
onUserVolumeStatsNotify	提示频道远端用户以及自己的音量回调
onNetworkTestQuality	会前网络探测质量上报回调
onNetworkTestResult	会前网络探测结果上报回调
onFirstRemoteVideoDecoded	远端用户第一帧解码成功回调
onFirstRemoteAuxiliaryStreamDecoded	收到第一帧远端辅流并解码成功回调
onAuthorizationExpired	签名过期回调
onNetworkQualityNotify	加入房间后的网络质量状态回调
onLocalAudioStateChangedNotify	本地音频流状态变化回调
onLocalVideoStateChangedNotify	本地视频流状态变化回调

接口	描述
onScreenShareStarted	辅流开启回调
onScreenShareStopped	辅流改变回调
onMediaConnectStateChangedNotify	媒体连接状态变化通知
onUserNameChangedNotify	本端修改昵称结果通知
onRemoteUserNameChangedNotify	远端用户昵称变更通知
onFirstLocalAudioFrame	本地音频首帧发送回调
onFirstLocalVideoFrameNotify	本地视频首帧渲染回调
onLocalVideoStatsNotify	本端视频统计回调
onRemoteVideoStatsNotify	远端视频统计回调
onLocalAudioStatsNotify	本端音频统计回调
onRemoteAudioStatsNotify	远端音频统计回调
onVideoResolutionChangedNotify	远端视频大小流改变回调
onStatsNotify	当前会话统计回调
onAudioMixStateChangedNotify	混音音频文件播放状态改变回调
onAudioFileFinished	音频文件播放结束回调
onAudioClipFinished	音效文件播放结束回调
onSeiSendMsgSuccess	音频SEI信息发送成功回调
onSeiRecvMsg	接收音频SEI信息回调
onStartPublishStream	开始旁路（RTMP）推流回调
onUpdateTransCoding	更新旁路（RTMP）推流消息
onStopPublishStream	停止旁路（RTMP）推流消息
onStreamPublishStateChange	RTMP推流状态回调
onLocalAudioMutedStatusDetected	本端禁音状态回调
onTopActiveSpeaker	远端音频最活跃用户回调
onRemoteMicrophoneStateChanged	远端麦克风设备状态变更通知
onUserNetworkQualityNotify	用户级网络质量回调
onMultiRoomMediaRelayStateChanged	跨房状态改变回调

onError

```
void onError(int error, String msg);
```

【功能说明】

错误回调。

【回调参数】

- error：错误码，具体请参见**HRTCErrorCode**。
- msg：错误描述。

onJoinRoomSuccess

```
void onJoinRoomSuccess(String roomId, String userId);
```

【功能说明】

成功加入房间回调。

【回调参数】

- roomId：房间ID。
- userId：用户ID。

onJoinRoomFailure

```
void onJoinRoomFailure(int error, String msg);
```

【功能说明】

加入房间失败回调。

【回调参数】

- error：错误码。
- msg：错误信息。

onLeaveRoom

```
void onLeaveRoom(HRTCLeaveReason reason, HRTCStatsInfo statsInfo);
```

【功能说明】

离开房间回调。

【回调参数】

- reason：自己离开房间的原因，具体请参见**HRTCLeaveReason**。
- statsInfo：会议期间的统计信息，具体请参见**HRTCStatsInfo**。

onRemoteUserOnline

```
void onRemoteUserOnline(String roomId, String userId, String userName);
```

【功能说明】

用户加入房间成功回调，不包括自己。

【回调参数】

- roomId：房间ID。
- userId：用户ID。
- userName：用户名称。

onRemoteUserOffline

```
void onRemoteUserOffline(String roomId, String userId, int reason);
```

【功能说明】

用户离开房间回调。

【回调参数】

- roomId：房间ID。
- userId：用户ID。
- reason：预留字段。

onConnectionChangedNotify

```
void onConnectionChangedNotify(HRTCConnStateTypes connStateTypes, HRTCConnChangeReason
connChangeReason, String description);
```

【功能说明】

网络连接状态改变。

【回调参数】

- connStateTypes：网络连接状态，具体请参见**HRTCConnStateTypes**。
- connChangeReason：网络连接状态原因，具体请参见**HRTCConnChangeReason**。
- description：描述。

onLogUploadResult

```
void onLogUploadResult(int result);
```

【功能说明】

日志上传结果回调。

【回调参数】

result：日志上传结果，0表示成功，1表示失败。

onLogUploadProgress

```
void onLogUploadProgress(int progress);
```

【功能说明】

日志上传进度回调。

【回调参数】

progress：进度，取值范围为[0,100]。

onUserRoleChangedNotify

```
void onUserRoleChangedNotify(String roomId, HRTCRoleType oldRole, HRTCRoleType newRole);
```

【功能说明】

用户角色改变。

【回调参数】

- roomId：发生角色切换的房间号。
- oldRole：改变前的角色，具体请参见**HRTCRoleType**。
- newRole：改变后的角色，具体请参见**HRTCRoleType**。

onAudioRouteStateChangedNotify

```
void onAudioRouteStateChangedNotify(HRTCAudioRoute route);
```

【功能说明】

音频路由改变。

【回调参数】

route：音频路由，具体请参见**HRTCAudioRoute**。

onUserAuxiliaryStreamAvailable

```
void onUserAuxiliaryStreamAvailable(String roomId, String userId, boolean available);
```

【功能说明】

辅流加入房间。

【回调参数】

- roomId：房间ID。
- userId：用户ID。
- available：true表示辅流开始推送，false表示辅流停止推送，提示用户关闭共享。

onRenderExternalVideoFrame

```
void onRenderExternalVideoFrame(String roomId, HRTCMediaDirection direction, String userId,
HRTCVideoFrame videoFrame);
```

【功能说明】

回调远端的视频原始数据流。

【回调参数】

- roomId：房间ID。
- userId：用户ID。
- videoFrame：视频数据。
- direction：数据源，本地数据，远端数据，具体请参见**HRTCMediaDirection**。

onPlaybackExternalAudioFrame

```
void onPlaybackExternalAudioFrame(String roomId,HRTCMediaDirection direction, HRTCAudioFrame audioFrame);
```

【功能说明】

回调远端的音频原始数据流。

【回调参数】

- roomId：房间ID。
- audioFrame：音频数据，具体请参见**HRTCAudioFrame**。
- direction：数据源，本地数据，远端数据，具体请参见**HRTCMediaDirection**。

onVideoStatsNotify

```
void onVideoStatsNotify(List<HRTCLocalVideoStats> localStats, List<HRTCRemoteVideoStats> remoteStats);
```

【功能说明】

视频流详情，2s触发一次回调。

【回调参数】

- localStats：本地视频发流详情参数，具体请参见**HRTCLocalVideoStats**。
- remoteStats：远端视频收流详情参数，具体请参见**HRTCRemoteVideoStats**。

⚠ 注意

只有本地一个用户入会时，不回调该方法。

onAudioStatsNotify

```
void onAudioStatsNotify(List<HRTCLocalAudioStats> localStats, List<HRTCRemoteAudioStats> remoteStats);
```

【功能说明】

音频流详情，2s触发一次回调。

【回调参数】

- localStats：本地音频发流详情，具体请参见**HRTCLocalAudioStats**。
- remoteStats：远端音频收流详情，具体请参见**HRTCRemoteAudioStats**。

onAuxiliaryStreamStatsNotify

```
void onAuxiliaryStreamStatsNotify(List<HRTCLocalVideoStats> localStats, List<HRTCRemoteVideoStats> remoteStats);
```

【功能说明】

辅流详情，2s触发一次回调。

【回调参数】

- localStats：本地辅流的发流详情，具体请参见**HRTCLocalVideoStats**。

- remoteStats：远端辅流的收流详情，具体请参见**HRTCRemoteVideoStats**。

onRemoteAudioStateChangedNotify

```
void onRemoteAudioStateChangedNotify(String userId, HRTCRemoteAudioStreamState state,
HRTCRemoteAudioStreamStateReason reason);
```

【功能说明】

远端音频流状态变化回调。

【回调参数】

- userId：远端用户ID。
- state：远端音频流状态，具体请参见**HRTCRemoteAudioStreamState**。
- reason：远端音频流状态变化原因，具体请参见**HRTCRemoteAudioStreamStateReason**。

onRemoteVideoStateChangedNotify

```
void onRemoteVideoStateChangedNotify(String userId, HRTCRemoteVideoStreamState state,
HRTCRemoteVideoStreamStateReason reason);
```

【功能说明】

远端视频流状态变化回调。

【回调参数】

- userId：远端用户ID。
- state：远端视频流状态，具体请参见**HRTCRemoteVideoStreamState**。
- reason：远端视频流状态变化原因，具体请参见**HRTCRemoteVideoStreamStateReason**。

onRejoinRoomSuccess

```
void onRejoinRoomSuccess(String roomId, String userId);
```

【功能说明】

重新加入房间回调。例如，网络异常后重连成功加入房间时触发。

【回调参数】

- roomId：房间ID。
- userId：用户ID。

onUserVolumeStatsNotify

```
void onUserVolumeStatsNotify(List<HRTCVolumeInfo> volumeInfos, int totalVolume);
```

【功能说明】

提示频道远端用户以及自己的音量回调。

【回调参数】

- volumeInfos：回调发言人信息列表，目前最多支持4人，包括自己，具体请参见**HRTCVolumeInfo**。

- totalVolume：远端混音后的总音量。

onNetworkTestQuality

```
void onNetworkTestQuality(HRTCNetworkQualityLevel level);
```

【功能说明】

会前网络探测质量上报，探测成功后每隔两秒刷新该level值。

【回调参数】

level：具体请参见**HRTCNetworkQualityLevel**。

onNetworkTestResult

```
void onNetworkTestResult(HRTCNetworkTestResult networkTestResult);
```

【功能说明】

会前网络探测结果上报，如果探测成功，该回调会在探测30s后上报，如果失败随时上报。

【回调参数】

networkTestResult：网络探测结果数据，具体请参见**HRTCNetworkTestResult**。

onFirstRemoteVideoDecoded

```
void onFirstRemoteVideoDecoded(String roomId, final String userId, int width, int height);
```

【功能说明】

远端用户第一帧解码成功回调。

【回调参数】

- roomId：房间ID。
- userId：用户ID。
- width：视频宽度。
- height：视频高度。

onFirstRemoteAuxiliaryStreamDecoded

```
void onFirstRemoteAuxiliaryStreamDecoded(String roomId, String userId, int width, int height);
```

【功能说明】

接收到第一帧远端辅流并解码成功，触发此回调。

【回调参数】

- roomId：视频流对应的房间ID。
- userId：视频流对应的用户ID。
- width：视频流宽。
- height：视频流高。

onAuthorizationExpired

```
void onAuthorizationExpired();
```

【功能说明】

签名过期回调，需要app调用**renewAuthorization**更新签名。

【回调参数】

无

onNetworkQualityNotify

```
void onNetworkQualityNotify(List<HRTCQualityInfo> upStreamQuality, List<HRTCQualityInfo>
downStreamQuality);
```

【功能说明】

加入房间后，基于流级别的网络质量状态回调。音频流、视频流分别回调。

【回调参数】

- 本端开始推流后，本端才会开始收到网络质量回调。
- upStreamQuality：上行网络质量状态，具体请参见**HRTCQualityInfo**。当前不可用。
- downStreamQuality：下行网络质量状态，具体请参见**HRTCQualityInfo**。

onLocalAudioStateChangedNotify

```
void onLocalAudioStateChangedNotify(HRTCLocalAudioStreamState state,
HRTCLocalAudioStreamStateReason reason);
```

【功能说明】

本地音频流状态变化回调。

【回调参数】

- state：本地音频流状态，具体请参见**HRTCLocalAudioStreamState**。
- reason：本地音频流状态变化原因，具体请参见**HRTCLocalAudioStreamStateReason**。

onLocalVideoStateChangedNotify

```
void onLocalVideoStateChangedNotify(HRTCLocalVideoStreamState state,
HRTCLocalVideoStreamStateReason reason);
```

【功能说明】

本地视频流状态变化回调。

【回调参数】

- state：本地视频流状态，具体请参见**HRTCLocalVideoStreamState**。
- reason：本地视频流状态变化原因，具体请参见**HRTCLocalVideoStreamStateReason**

onScreenShareStarted

```
void onScreenShareStarted();
```

【功能说明】

屏幕流共享开启，触发此回调。

onScreenShareStopped

```
void onScreenShareStopped(int reason)
```

【功能说明】

屏幕流共享状态改变，触发此回调。

【回调参数】

reason：屏幕共享状态改变原因。

注意

- SDK 1.7.1及以上版本支持。
- reason：预留参数。

onMediaConnectStateChangedNotify

```
void onMediaConnectStateChangedNotify(HRTCMediaConnStateTypes state,
HRTCMediaConnChangeReason reason, String description);
```

【功能说明】

媒体服务器连接状态变更通知。

【回调参数】

- state：连接状态，具体请参见**HRTCMediaConnStateTypes**。
- reason：连接状态变化的原因，具体请参见**HRTCMediaConnChangeReason**。
- description：连接状态变化原因描述。

注意

加入房间过后，收到媒体服务的数据包时，返回Connected消息，超过6s没有收到包，则返回Failed消息。

onUserNameChangedNotify

```
void onUserNameChangedNotify(String oldUserName, String newUserName);
```

【功能说明】

调用changeUserName接口成功后，上报此事件通知修改昵称结果。

【回调参数】

- oldUserName：修改前昵称。
- newUserName：修改后昵称。

onRemoteUserNameChangedNotify

```
void onRemoteUserNameChangedNotify(String roomId，String userId, String userName);
```

【功能说明】

远端用户修改昵称后，通知本端昵称变更。

【回调参数】

- userId：修改昵称的用户id。
- userName：修改后的昵称。
- roomId：修改昵称的房间。

onFirstLocalAudioFrame

```
void onFirstLocalAudioFrame(int elapsed);
```

【功能说明】

本地音频首帧发送回调。

【回调参数】

elapsed：从入会到本地音频首帧发送所用的时间，单位为ms。

onFirstLocalVideoFrameNotify

```
void onFirstLocalVideoFrameNotify(int elapsed);
```

【功能说明】

本地视频首帧渲染回调。

【回调参数】

elapsed：从开始采集到本地视频首帧渲染所用的时间，单位为ms。

onLocalVideoStatsNotify

```
void onLocalVideoStatsNotify(List<HRTCLocalVideoStats> localStats);
```

【功能说明】

本端视频流详情，2s触发一次回调。

【回调参数】

localStats：本地视频发流详情参数，具体请参见**HRTCLocalVideoStats**。

onRemoteVideoStatsNotify

```
void onRemoteVideoStatsNotify(List<HRTCRemoteVideoStats> remoteStats);
```

【功能说明】

远端视频流详情，2s触发一次回调。

【回调参数】

remoteStats：远端视频收流详情参数，具体请参见**HRTCRemoteVideoStats**。

onLocalAudioStatsNotify

```
void onLocalAudioStatsNotify(List<HRTCLocalAudioStats> localStats);
```

【功能说明】

本端音频流详情，2s触发一次回调。

【回调参数】

localStats：本地音频发流详情，具体请参见**HRTCLocalAudioStats**。

onRemoteAudioStatsNotify

```
void onRemoteAudioStatsNotify(List<HRTCRemoteAudioStats> remoteStats);
```

【功能说明】

远端音频流详情，2s触发一次回调。

【回调参数】

remoteStats：远端音频收流详情，具体请参见**HRTCRemoteAudioStats**。

onVideoResolutionChangedNotify

```
void onVideoResolutionChangedNotify(String userid, int width, int height);
```

【功能说明】

远端视频大小改变回调。

【回调参数】

- userId：用户ID。
- width：视频流宽度。
- height：视频流高度。

onStatsNotify

```
void onStatsNotify(List<HRTCOnStats> hrtcOnStats);
```

【功能说明】

当前会话统计回调。

【回调参数】

hrtcOnStats：会话统计信息，具体请参见**HRTCOnStats**。

onAudioMixStateChangedNotify

```
void onAudioMixStateChangedNotify(HRTCAudioFileState state, HRTCAudioFileReason reason, long value);
```

【功能说明】

音频文件播放状态改变回调。

【回调参数】

- state：音频文件播放状态，具体请参见**HRTCAudioFileState**。
- reason：音频文件播放状态变化原因，具体请参见**HRTCAudioFileReason**。

- value：state为HRTC_AUDIO_FILE_OPEN_COMPLETED表示音频文件的时长，单位为ms；state为HRTC_AUDIO_FILE_POSITION_UPDATE表示当前播放的位置，单位为ms。其他情况下，value值无意义。

onAudioFileFinished

```
void onAudioFileFinished();
```

【功能说明】

音频文件播放结束回调。

【回调参数】

无

onAudioClipFinished

```
void onAudioClipFinished(int soundId);
```

【功能说明】

音效文件播放结束回调。

【回调参数】

soundId：音效ID，取值>= 0。

onSeiSendMsgSuccess

```
void onSeiSendMsgSuccess(String message);
```

【功能说明】

音频SEI信息发送成功回调。

【回调参数】

message：发送SEI信息的内容。

onSeiRecvMsg

```
void onSeiRecvMsg(String userId, String message);
```

【功能说明】

接收音频SEI信息回调。

【回调参数】

- userId：用户ID。
- message：接收SEI信息的内容。

onError

```
virtual void onError(IHRTCConnection* conn, int error, const char* msg)
```

【功能说明】

发生错误，触发此回调。返回客户端错误码或者服务端错误码。

【回调参数】

- conn：连接对象。
- error：错误码，具体请参见**HRTCErrorCode**。
- msg：错误描述。

onConnectionChangedNotify

```
virtual void onConnectionChangedNotify(IHRTCConnection* conn, HRTCConnStateType state,
HRTCConnChangeReason reason, const char* description)
```

【功能说明】

连接状态改变回调。

【回调参数】

- conn：连接对象。
- state：连接状态类型，具体请参见**HRTCConnStateTypes**。
- reason：连接状态改变原因，具体请参见**HRTCConnChangeReason**。
- description：连接状态改变描述。

onAuthorizationExpired

```
virtual void onAuthorizationExpired(IHRTCConnection* conn);
```

【功能说明】

鉴权签名过期回调，需要app调用**renewAuthorization**更新签名。

【回调参数】

conn：连接对象。

onJoinRoomSuccess

```
virtual void onJoinRoomSuccess(IHRTCConnection* conn, const char* userId)
```

【功能说明】

成功加入房间，触发此回调。

【回调参数】

- conn：连接对象。
- userId：新加入房间的用户ID。

onJoinRoomFailure

```
virtual void onJoinRoomFailure(IHRTCConnection* conn, int error, const char* msg)
```

【功能说明】

加入房间失败，触发此回调。

【回调参数】

- conn：连接对象。
- error：错误码，具体请参见**HRTCErrorCode**。
- msg：错误描述。

onRejoinRoomSuccess

```
virtual void onRejoinRoomSuccess(IHRTCConnection* conn, const char* userId)
```

【功能说明】

重新加入房间回调。例如，网络异常后重连成功加入房间触发。

【回调参数】

- conn：连接对象。
- userId：用户ID。

onLeaveRoom

```
virtual void onLeaveRoom(IHRTCConnection* conn, HRTCLeaveReason reason, const HRTCStatsInfo &statsInfo)
```

【功能说明】

离开房间，触发此回调。

【回调参数】

- conn：连接对象。
- reason：离开的房间原因，具体请参见**HRTCLeaveReason**。
- statsInfo：卡顿统计信息，具体请参见**HRTCStatsInfo**。

onUserRoleChangedNotify

```
virtual void onUserRoleChangedNotify(IHRTCConnection* conn, HRTCRoleType oldRole, HRTCRoleType newRole)
```

【功能说明】

用户角色切换成功，触发此回调。

【回调参数】

- conn：连接对象。
- oldRole：切换前的角色。具体请参见**HRTCRoleType**。
- newRole：切换成功后的角色。具体请参见**HRTCRoleType**。

onRemoteUserOnline

```
virtual void onRemoteUserOnline(IHRTCConnection* conn, const char* userId, const char* userName)
```

【功能说明】

远端joiner用户加入当前房间，触发此回调。该回调提示有远端joiner用户加入了房间，并返回新加入用户的ID。

【回调参数】

- conn：连接对象。
- userId：远端用户ID。
- userName：远端用户昵称。

onRemoteUserOffline

```
virtual void onRemoteUserOffline(IHRTCConnection* conn, const char* userId, int reason)
```

【功能说明】

远端joiner用户离开当前房间，触发此回调。

本端用户离开当前房间，会回调当前房间所有用户offline。

【回调参数】

- conn：连接对象。
- userId：离开房间的远端用户ID。
- reason：远端用户离线原因，预留参数。

onRemoteUserNameChangedNotify

```
virtual void onRemoteUserNameChangedNotify(IHRTCConnection* conn, const char* userId, const char* userName)
```

【功能说明】

远端用户昵称变化，触发此回调。

【回调参数】

- conn：连接对象。
- userId：用户ID。
- userName：变更后的昵称。

onUserNameChangedNotify

```
virtual void onUserNameChangedNotify(IHRTCConnection* conn, const char* oldUserName, const char* newUserName)
```

【功能说明】

本端用户昵称变化，触发此回调。

【回调参数】

- conn：连接对象。
- oldUserName：变更前的昵称。
- newUserName：变更后的昵称。

onRemoteAudioStateChangedNotify

```
virtual void onRemoteAudioStateChangedNotify(IHRTCConnection* conn, const char* userId, HRTCRemoteAudioStreamState state, HRTCRemoteAudioStreamStateReason reason)
```

【功能说明】

远端音频流状态变化回调。

【回调参数】

- conn：连接对象。
- userId：远端用户ID。

- state：远端音频流状态，具体请参见**HRTCRemoteAudioStreamState**。
- reason：远端音频流状态变化原因，具体请参见**HRTCRemoteAudioStreamStateReason**。

onRemoteVideoStateChangedNotify

```
virtual void onRemoteVideoStateChangedNotify(IHRTCConnection* conn, const char* userId,
HRTCRemoteVideoStreamState state, HRTCRemoteVideoStreamStateReason reason)
```

【功能说明】

远端视频流状态变化回调。

【回调参数】

- conn：连接对象。
- userId：远端用户ID。
- state：远端视频流状态，具体请参见**HRTCRemoteVideoStreamState**。
- reason：远端视频流状态变化原因，具体请参见**HRTCRemoteVideoStreamStateReason**。

onUserVolumeStatsNotify

```
virtual void onUserVolumeStatsNotify(IHRTCConnection* conn, const HRTCVolumeInfo* userVolumes,
unsigned int userVolumesCount, unsigned int totalVolume)
```

【功能说明】

用户音量状态回调。通过**enableUserVolumeNotify**开启并设置回调周期，定时上报。

【回调参数】

- conn：连接对象。
- userVolumes：HRTCVolumeInfo，userId，volume。
- userVolumesCount：音量上报用户数组的大小。
- totalVolume：总音量。

onUserAuxiliaryStreamAvailable

```
virtual void onUserAuxiliaryStreamAvailable(IHRTCConnection* conn, const char* userId, bool available)
```

【功能说明】

远端开启，停止辅流后，触发此回调。

【回调参数】

- conn：连接对象。
- userId：远端用户ID。
- available：true表示远端开启推辅流，false表示远端停止辅流。

onFirstRemoteVideoDecoded

```
virtual void onFirstRemoteVideoDecoded(IHRTCConnection* conn, const char* userId, int width, int height)
```

【功能说明】

远端用户视频流第一帧解码成功，触发此回调。

【回调参数】

- conn：连接对象。
- userId：用户ID。
- width：视频宽。
- height：视频高。

onFirstRemoteAuxiliaryStreamDecoded

```
virtual void onFirstRemoteAuxiliaryStreamDecoded(IHRTCConnection* conn, const char* userId, int width, int height)
```

【功能说明】

远端用户辅流第一帧解码成功，触发此回调。

【回调参数】

- conn：连接对象。
- userId：用户ID。
- width：视频宽。
- height：视频高。

onAudioStatsNotify

```
virtual void onAudioStatsNotify(IHRTCConnection* conn, HRTCLocalAudioStats *localStats, unsigned int localStatsCount, HRTCRemoteAudioStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

音频流详情，2s触发一次回调。

【回调参数】

- conn：连接对象。
- localStats：本地音频发流统计，具体请参见**HRTCLocalAudioStats**。
- localStatsCount：localStats数组长度。
- remoteStats：远端音频收流详情，具体请参见**HRTCRemoteAudioStats**。
- remoteStatsCount：remoteStats数组长度。

⚠ 注意

- 当无本地音频时，localStatsCount为0，localStats为空指针。
- 当无远端音频时，remoteStatsCount为0，remoteStats为空指针。

onVideoStatsNotify

```
virtual void onVideoStatsNotify(IHRTCConnection* conn, HRTCLocalVideoStats *localStats, unsigned int localStatsCount, HRTCRemoteVideoStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

视频流详情，2s触发一次回调。

【回调参数】

- conn：连接对象。
- localStats：本地视频发流统计，具体请参见**HRTCLocalVideoStats**。
- localStatsCount：localStats数组长度。
- remoteStats：远端视频收流统计，具体请参见**HRTCRemoteVideoStats**。
- remoteStatsCount：remoteStats数组长度。

⚠ 注意

- 当无本地视频时，localStatsCount为0，localStats为空指针。
- 当无远端视频时，remoteStatsCount为0，remoteStats为空指针。

onAuxiliaryStreamStatsNotify

```
virtual void onAuxiliaryStreamStatsNotify(IHRTCConnection* conn, HRTCLocalVideoStats *localStats,
unsigned int localStatsCount, HRTCRemoteVideoStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

辅流详情，2s触发一次回调。

【回调参数】

- conn：连接对象。
- localStats：本地辅流发流详情，具体请参见**HRTCLocalVideoStats**。
- localStatsCount：本地辅流发流数量。
- remoteStats：远端辅流收流详情，具体请参见**HRTCRemoteVideoStats**。
- remoteStatsCount：远端辅流收流数量。

onNetworkQualityNotify

```
virtual void onNetworkQualityNotify(IHRTCConnection* conn, HRTCQualityInfo* localQuality, unsigned int
localQualityCount, HRTCQualityInfo* remoteQuality, unsigned int remoteQualityCount)
```

【功能说明】

房间内客户端基于流级别的网络质量实时上报，默认开启，每2s上报一次，有数据流时才会回调，音频流、视频流分开回调。

【回调参数】

- conn：连接对象。
- localQuality：本地上行网络质量，该参数暂时不使用。
- localQualityCount：正在上报的网络质量数量，该参数暂时不使用。
- remoteQualityCount：正在上报的流的数量，集合的大小。

onError

```
virtual void onError(int error, const char* msg)
```

【功能说明】

发生错误，触发此回调。返回**客户端错误码**或者**服务端错误码**。

【回调参数】

- error：错误码，具体请参见**HRTCErrorCode**。
- msg：错误描述。

onJoinRoomSuccess

```
virtual void onJoinRoomSuccess(const char* roomId, const char* userId)
```

【功能说明】

成功加入房间，触发此回调。

【回调参数】

- roomId：新加入的房间ID。
- userId：新加入房间的用户ID。

onJoinRoomFailure

```
virtual void onJoinRoomFailure(int error, const char* msg)
```

【功能说明】

加入房间失败，触发此回调。

【回调参数】

- error：错误码，具体请参见**HRTCErrorCode**。
- msg：错误描述。

onLeaveRoom

```
virtual void onLeaveRoom(HRTCLeaveReason reason, const HRTCStatsInfo &statsInfo)
```

【功能说明】

离开房间，触发此回调。

【回调参数】

- reason：离开的房间原因，具体请参见**HRTCLeaveReason**。
- statsInfo：卡顿统计信息，具体请参见**HRTCStatsInfo**。

⚠ 注意

APP调用**leaveRoom**接口时，会返回**HRTC_LEAVE_REASON_USER_LEAVE_ROOM**，可以通过以下任一方式回退到登录界面。

- APP在调用**leaveRoom**接口时退到登录界面，或者在收到onLeaveRoom回调，且回调消息不等于**HRTC_LEAVE_REASON_USER_LEAVE_ROOM**时（防止重复操作）退到登录界面。
- APP只在收到onLeaveRoom消息时退到登录界面。

onRemoteUserOnline

```
virtual void onRemoteUserOnline(const char* roomId, const char* userId, const char* userName)
```

【功能说明】

远端joiner用户加入当前房间，触发此回调。该回调提示有远端joiner用户加入了房间，并返回新加入用户的ID。

【回调参数】

- roomId：房间ID。
- userId：远端用户ID。
- userName：远端用户昵称。

onRemoteUserOffline

```
virtual void onRemoteUserOffline(const char* roomId, const char* userId, int reason)
```

【功能说明】

远端joiner用户离开当前房间，触发此回调。

本端用户离开当前房间，会回调当前房间所有用户offline。

【回调参数】

- roomId：当前房间的房间ID。
- userId：离开房间的远端用户ID。
- reason：远端用户离线原因，预留参数。

onRemoteUserNameChangedNotify

```
virtual void onRemoteUserNameChangedNotify(const char* roomId, const char* userId, const char* userName)
```

【功能说明】

远端用户昵称变化，触发此回调。

【回调参数】

- roomId：房间ID。
- userId：用户ID。
- userName：变更后的昵称。

onUserNameChangedNotify

```
virtual void onUserNameChangedNotify(const char* oldUserName, const char* newUserName)
```

【功能说明】

本端用户昵称变化，触发此回调。

【回调参数】

- oldUserName：变更前的昵称。
- newUserName：变更后的昵称。

onFirstRemoteAuxiliaryStreamDecoded

```
virtual void onFirstRemoteAuxiliaryStreamDecoded(const char* roomId, const char* userId, int width, int height)
```

【功能说明】

接收到第一帧远端辅流并解码成功，触发此回调。

【回调参数】

- roomId：视频流对应的房间ID。
- userId：视频流对应的用户ID。
- width：视频流宽。
- height：视频流高。

onFirstRemoteVideoDecoded

```
virtual void onFirstRemoteVideoDecoded(const char* roomId, const char* userId, int width, int height)
```

【功能说明】

接收到第一帧远端视频流并解码成功，触发此回调。

【回调参数】

- roomId：视频流对应的房间ID。
- userId：视频流对应的用户ID。
- width：视频流宽。
- height：视频流高。

onConnectionChangedNotify

```
virtual void onConnectionChangedNotify(HRTCConnStateTypes connType, HRTCConnChangeReason reason, const char* description)
```

【功能说明】

网络连接状态发生变化，触发此回调。

【回调参数】

- connType：网络连接状态。具体请参见**HRTCConnStateTypes**。
- reason：网络连接状态发生变化原因。具体请参见**HRTCConnChangeReason**。
- description：错误原因描述。

onDeviceStateChangedNotify

```
virtual void onDeviceStateChangedNotify(const char* deviceId, HRTCDeviceType deviceType,
HRTCDeviceState deviceState)
```

【功能说明】

设备状态发生变化，触发此回调。

【回调参数】

- deviceId：系统设备标识，如系统音频播放设备标识可通过**getPlaybackDevices**获取。
- deviceType：系统设备类型，具体请参见**HRTCDeviceType**。
- deviceState：系统设备状态，具体请参见**HRTCDeviceState**。

⚠ 注意

通话前插拔设备会上报变化。

onDeviceVolumeChangedNotify

```
virtual void onDeviceVolumeChangedNotify(HRTCDeviceType deviceType, unsigned int volume, unsigned int
muted)
```

【功能说明】

音频设备音量发生变化，触发此回调。

【回调参数】

- deviceType：系统设备类型，具体请参见**HRTCDeviceType**。
- volume：音量。
- muted：true表示设备静音，false表示设备未静音。

⚠ 注意

通话前调整音频设备音量和静音会上报变化。

onLogUploadResult

```
virtual void onLogUploadResult(int result)
```

【功能说明】

日志上传结果回调。

【回调参数】

result：日志上传结果。

- 0：上传成功
- >0：上传失败

onLogUploadProgress

```
virtual void onLogUploadProgress(int progress)
```

【功能说明】

日志上传进度回调。

【回调参数】

progress：日志上传进度。取值范围为[0,100]。

onUserRoleChangedNotify

```
virtual void onUserRoleChangedNotify(const char* roomId, HRTCRoleType oldRole, HRTCRoleType newRole)
```

【功能说明】

用户角色切换成功，触发此回调。

【回调参数】

- roomId：发生角色切换的房间号。
- oldRole：切换前的角色。具体请参见**HRTCRoleType**。
- newRole：切换成功后的角色。具体请参见**HRTCRoleType**。

onScreenShareStarted

```
virtual void onScreenShareStarted()
```

【功能说明】

屏幕流共享开启，触发此回调。

onScreenShareStopped(int reason)

```
virtual void onScreenShareStopped(int reason)
```

【功能说明】

屏幕流共享状态改变，触发此回调。

【回调参数】

reason：屏幕共享改变原因。

onUserAuxiliaryStreamAvailable

```
virtual void onUserAuxiliaryStreamAvailable(const char* roomId, const char* userId, bool available)
```

【功能说明】

远端开启，停止辅流后，触发此回调。

【回调参数】

- roomId：房间ID。
- userId：远端用户ID。
- available：true表示远端开启推辅流，false表示远端停止辅流。

onVideoStatsNotify

```
virtual void onVideoStatsNotify(HRTCLocalVideoStats *localStats, unsigned int localStatsCount,
HRTCRemoteVideoStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

视频流详情，2s触发一次回调。

【回调参数】

- localStats：本地视频发流统计，具体请参见**HRTCLocalVideoStats**。
- localStatsCount：localStats数组长度。
- remoteStats：远端视频收流统计，具体请参见**HRTCRemoteVideoStats**。
- remoteStatsCount：remoteStats数组长度。

⚠ 注意

- 当无本地视频时，localStatsCount为0，localStats为空指针，需要先判断不为空再使用，否则可能引发空指针错误。
- 当无远端视频时，remoteStatsCount为0，remoteStats为空指针，需要先判断不为空再使用，否则可能引发空指针错误。

onAudioStatsNotify

```
virtual void onAudioStatsNotify(HRTCLocalAudioStats *localStats, unsigned int localStatsCount,
HRTCRemoteAudioStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

音频流详情，2s触发一次回调。

【回调参数】

- localStats：本地音频发流统计，具体请参见**HRTCLocalAudioStats**。
- localStatsCount：localStats数组长度。
- remoteStats：远端音频收流统计，具体请参见**HRTCRemoteAudioStats**。
- remoteStatsCount：remoteStats数组长度。

⚠ 注意

- 当无本地音频时，localStatsCount为0，localStats为空指针，需要先判断不为空再使用，否则可能引发空指针错误。
- 当无远端音频时，remoteStatsCount为0，remoteStats为空指针，需要先判断不为空再使用，否则可能引发空指针错误。

onAuxiliaryStreamStatsNotify

```
virtual void onAuxiliaryStreamStatsNotify(HRTCLocalVideoStats *localStats, unsigned int localStatsCount,
HRTCRemoteVideoStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

辅流详情，2s触发一次回调。

【回调参数】

- localStats：本地辅流发流统计，具体请参见**HRTCLocalVideoStats**。
- localStatsCount：localStats数组长度。
- remoteStats：远端辅流收流统计，具体请参见**HRTCRemoteVideoStats**。
- remoteStatsCount：remoteStats数组长度。

⚠ 注意

- 当无本地共享频时，localStatsCount为0，localStats为空指针，需要先判断不为空再使用，否则可能引发空指针错误。
- 当无远端共享频时，remoteStatsCount为0，remoteStats为空指针，需要先判断不为空再使用，否则可能引发空指针错误。

onAuthorizationExpired

```
virtual void onAuthorizationExpired();
```

【功能说明】

鉴权签名过期回调，需要app调用更新签名。

onRemoteAudioStateChangedNotify

```
virtual void onRemoteAudioStateChangedNotify(const char* userId, HRTCRemoteAudioStreamState state,
HRTCRemoteAudioStreamStateReason reason)
```

【功能说明】

远端音频流状态变化回调。

【回调参数】

- userId：远端用户ID。
- state：远端音频流状态，具体请参见**HRTCRemoteAudioStreamState**。
- reason：远端音频流状态变化原因，具体请参见**HRTCRemoteAudioStreamStateReason**。

onRemoteVideoStateChangedNotify

```
virtual void onRemoteVideoStateChangedNotify(const char* userId, HRTCRemoteVideoStreamState state,
HRTCRemoteVideoStreamStateReason reason)
```

【功能说明】

远端视频流状态变化回调。

【回调参数】

- userId：远端用户ID。
- state：远端视频流状态，具体请参见**HRTCRemoteVideoStreamState**。
- reason：远端视频流状态变化原因，具体请参见**HRTCRemoteVideoStreamStateReason**。

onRenderAuxiliaryExternalVideoFrame

```
virtual void onRenderAuxiliaryExternalVideoFrame(const char* roomId, HRTCMediaDirection direction, const char* userId, HRTCVideoFrame& videoFrame)
```

【功能说明】

辅流自渲染回调。需要调用setAuxiliaryExternalVideoFrameOutput接口开启辅流自渲染，从而触发该回调。

【回调参数】

- roomId：房间ID。
- direction：数据源，本地数据，远端数据，具体请参见**HRTCMediaDirection**。
- userId：用户ID。
- videoFrame：辅流详情，具体请参见**HRTCVideoFrame**。

onRejoinRoomSuccess

```
virtual void onRejoinRoomSuccess(const char* roomId, const char* userId)
```

【功能说明】

重新加入房间回调。例如网络异常后重连成功加入房间触发。

【回调参数】

- roomId：房间ID。
- userId：用户ID。

onNetworkTestQuality

```
virtual void onNetworkTestQuality(HRTCNetworkQualityLevel level)
```

【功能说明】

加房间前网络探测回调。

【回调参数】

level：网络质量，具体请参见**HRTCNetworkQualityLevel**。

onNetworkTestResult

```
virtual void onNetworkTestResult(HRTCNetworkTestResult& networkTestResult)
```

【功能说明】

加房间前网络探测结果回调。

【回调参数】

networkTestResult：主要包括测试成功与否、上行和下行的网络带宽、丢包、延时和抖动，具体请参见**HRTCNetworkTestResult**。

onUserVolumeStatsNotify

```
virtual void onUserVolumeStatsNotify(const HRTCVolumeInfo* userVolumes, unsigned int userVolumesCount, unsigned int totalVolume)
```

【功能说明】

用户音量状态回调。通过**enableUserVolumeNotify**开启并设置回调周期，定时上报。

【回调参数】

- userVolumes：用户信息，具体请参见**HRTCVolumeInfo**。
- userVolumesCount：上报的用户人数，包含本地用户。
- totalVolume：总音量。

onTopActiveSpeaker

```
virtual void onTopActiveSpeaker(const char* userId, bool noStream)
```

【功能说明】

声控画面的用户ID变化时，触发此回调。该回调主要用于0号会场场景。

【回调参数】

userId：返回当前声控画面的用户ID。

noStream：该用户是否有视频流。

⚠ 注意

0号会场模式下，SDK会持续监测（根据一定时间内用户音量大小）当前最活跃的用户，如果最活跃用户发生变化，则触发此回调并上报当前最活跃的用户userId。

onLocalAudioStateChangedNotify

```
virtual void onLocalAudioStateChangedNotify(HRTCLocalAudioStreamState state,
HRTCLocalAudioStreamStateReason reason)
```

【功能说明】

本地音频状态改变，触发此回调。

【回调参数】

- state：本地音频状态，具体请参见**HRTCLocalAudioStreamState**。
- reason：本地音频状态改变的原因，具体请参见**HRTCLocalAudioStreamStateReason**。

onLocalVideoStateChangedNotify

```
virtual void onLocalVideoStateChangedNotify(HRTCLocalVideoStreamState state,
HRTCLocalVideoStreamStateReason reason)
```

【功能说明】

本地视频状态改变，触发此回调。

【回调参数】

- state：本地视频状态，具体请参见**HRTCLocalVideoStreamState**。
- reason：本地视频状态改变原因，具体请参见**HRTCLocalVideoStreamStateReason**。

onMediaConnectStateChangedNotify

```
virtual void onMediaConnectStateChangedNotify(HRTCMediaConnStateTypes state,
HRTCMediaConnChangeReason reason, const char* description)
```

【功能说明】

媒体服务器连接状态变更通知。

【回调参数】

- state：与媒体服务器连接状态，具体请参见**HRTCMediaConnStateTypes**。
- reason：连接状态变化的原因，具体请参见**HRTCMediaConnChangeReason**。
- description：连接状态变化原因描述。

⚠ 注意

加入房间过后，收到媒体服务的数据包时，返回Connected消息，超过6s没有收到包，则返回Failed消息。

onStartAllRemoteViewResult

```
virtual void onStartAllRemoteViewResult(int errCode, const char* errMsg, unsigned int counts, const
HRTCSetupRemoteViewResult* results)
```

【功能说明】

批量选看结果回调。

【回调参数】

- errCode：错误码。
- errMsg：错误信息。
- counts：results数组大小。
- results：批量选看结果，具体请参见**HRTCSetupRemoteViewResult**。

onStatsNotify

```
virtual void onStatsNotify(HRTCOnStats *rtcStats)
```

【功能说明】

当前会话统计回调。

【回调参数】

rtcStats：当前会话统计，具体请参见**HRTCOnStats**。

onLocalVideoStatsNotify

```
virtual void onLocalVideoStatsNotify(const HRTCLocalVideoStats *localStats, unsigned int localStatsCount)
```

【功能说明】

本地视频流详情，2s触发一次回调。

【回调参数】

- localStats：本地视频收流统计，具体请参见**HRTCLocalVideoStats**。
- localStatsCount：localStats数组长度。

onRemoteVideoStatsNotify

```
virtual void onRemoteVideoStatsNotify(const HRTCRemoteVideoStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

远端视频流详情，2s触发一次回调。

【回调参数】

- remoteStats：远端视频收流统计，具体请参见**HRTCRemoteVideoStats**。
- remoteStatsCount：remoteStats数组长度。

onLocalAudioStatsNotify

```
virtual void onLocalAudioStatsNotify(const HRTCLocalAudioStats *localStats, unsigned int localStatsCount)
```

【功能说明】

本地音频流详情，2s触发一次回调。

【回调参数】

- localStats：本地音频收流统计，具体请参见**HRTCLocalAudioStats**。
- localStatsCount：localStats数组长度。

onRemoteAudioStatsNotify

```
virtual void onRemoteAudioStatsNotify(const HRTCRemoteAudioStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

远端音频流详情，2s触发一次回调。

【回调参数】

- remoteStats：远端音频收流统计，具体请参见**HRTCRemoteAudioStats**。
- remoteStatsCount：remoteStats数组长度。

onVideoResolutionChangedNotify

```
virtual void onVideoResolutionChangedNotify(const char* userId, int width, int height)
```

【功能说明】

远端视频分辨率大小改变，触发此回调。

【回调参数】

- userId：用户ID。
- width：视频分辨率改变后的宽。
- height：视频分辨率改变后的高。

onAudioClipFinished

```
virtual void onAudioClipFinished(int soundId)
```

【功能说明】

音效文件播放结束，触发此回调。

【回调参数】

soundId：音效ID，取值>=0。

onAudioFileFinished

```
virtual void onAudioFileFinished()
```

【功能说明】

音频文件播放结束，触发此回调。

【回调参数】

无

onAudioMixStateChangedNotify

```
virtual void onAudioMixStateChangedNotify(HRTCAudioFileState state, HRTCAudioFileReason reason,
unsigned long long value)
```

【功能说明】

音频文件播放状态改变，触发此回调。

【回调参数】

- state：音频播放状态，具体请参见**HRTCAudioFileState**。
- reason：音频播放状态改变原因，具体请参见**HRTCAudioFileReason**。
- value：state为HWRtcAudioFileOpenCompleted表示音频文件的时长，单位为ms；state为HWRtcAudioFilePositionUpdate表示当前播放的位置，单位为ms。其他情况下，value值无意义。

onSeiSendMsgSuccess

```
void onSeiSendMsgSuccess(const char* message);
```

【功能说明】

音频SEI信息发送成功回调。

回调参数

message：发送SEI信息的内容。

onSeiRecvMsg

```
void onSeiRecvMsg(const char* userId, const char* message);
```

【功能说明】

接收音频SEI信息回调。

回调参数

- userId：用户ID。
- message：接收SEI信息的内容。

onStartPublishStream

```
void onStartPublishStream(int code, const char* taskId);
```

【功能说明】

开始旁路（RTMP）推流回调。

回调参数

- code：错误码，成功为0，失败请参考**错误码**。
- taskId：任务Id。

onUpdateTransCoding

```
void onUpdateTransCoding(int code, const char* taskId);
```

【功能说明】

更新旁路（RTMP）推流消息。

回调参数

- code：错误码，成功为0，失败请参考**错误码**。
- taskId：任务Id。

onStopPublishStream

```
void onStopPublishStream(int code, const char* taskId);
```

【功能说明】

停止旁路（RTMP）推流消息。

回调参数

- code：错误码，成功为0，失败请参考**错误码**。
- taskId：任务Id。

onStreamPublishStateChange

```
void onStreamPublishStateChange(int code, const char* taskId, const HRTCUrlStatusList * urlStatu);
```

【功能说明】

RTMP推流状态回调。

回调参数

- code：错误码，成功为0，失败请参考**错误码**。
- taskId：任务Id。
- urlStatu：推流的url状态，具体请参见**HRTCUrlStatusList**。

onLocalAudioMutedStatusDetected

```
void onLocalAudioMutedStatusDetected();
```

【功能说明】

本端禁音检测回调。通过enableUserVolumeNotify开启并设置回调周期，禁音检测以6次音量回调为一个上报周期，**enableUserVolumeNotify**的参数建议设置成200（即最快1200ms触发禁音检测回调一次）。

【回调参数】

无

onTopActiveSpeaker

```
public void onTopActiveSpeaker(String userId, boolean noStream);
```

【功能说明】

上报当前最活跃的用户userId。该回调主要用于0号会场场景（额外订阅uid为0的用户音频）。

【回调参数】

userId：返回当前远端音量最活跃的用户ID。

noStream：该用户是否有视频流。

⚠ 注意

0号会场模式下，SDK会持续监测（根据一定时间内用户音量大小）当前最活跃的用户，如果最活跃用户发生变化，则触发此回调并上报当前最活跃的用户userId。

onRemoteMicrophoneStateChanged

```
void onRemoteMicrophoneStateChanged(String userId, HRTCRemoteMicState state);
```

【功能说明】

远端麦克风设备状态变更回调。

【回调参数】

- userId：用户id。
- state：麦克风设备状态，具体请参见**HRTCRemoteMicState**。

onUserNetworkQualityNotify

```
void onUserNetworkQualityNotify(String roomId, String userId, HRTCNetworkQualityLevel txQuality,
HRTCNetworkQualityLevel rxQuality);
```

【功能说明】

支持用户上报各自与近端SFU间的上下行网络质量，基于用户级，使本地用户能获取同房间内远端用户与其近端SFU间的网络质量。CMD广播时为房间级，将广播给订阅了此主播流的用户或者此主播被选为TOPN用户且观众订阅了该TOPN用户。

【回调参数】

- roomId：用户所在房间号。
- userId：上报的用户id，0为本地，非0为远端。

- txQuality：该用户的上行网络质量，具体请参见**HRTCNetworkQualityLevel**。
- rxQuality：该用户的下行网络质量，具体请参见**HRTCNetworkQualityLevel**。

⚠ 注意

- 此接口不支持跨房场景、WebRTC场景。
- 不支持RTSA。

onMultiRoomMediaRelayStateChanged

```
void onMultiRoomMediaRelayStateChanged(const char *roomId, HRTCMultiRoomMediaRelayState state,
HRTCMultiRoomMediaRelayStateCode code);
```

【功能说明】

跨房状态回调。

【回调参数】

- roomId：跨房房间号。
- state：状态类型，具体请参见**HRTCMultiRoomMediaRelayState**。
- code：状态的具体原因，具体请参见**HRTCMultiRoomMediaRelayStateCode**。

5.4.3 HRTCConnection

5.4.3.1 接口总览

本章节介绍了Android SDK的HRTCConnection接口详情。

HRTCConnection按照其功能可分类为：初始化等基础接口、房间功能、音频管理、视频管理、屏幕共享。

说明

单击下图中相应接口名称，可快速跳转到相应接口位置查看其使用方法。

表 5-14 初始化等基础接口

接口	描述
release	释放Connection
setEncryption	设置端到端加密模式
setNetworkBandwidth	设置网络带宽限制

表 5-15 房间功能接口

接口	描述
joinRoom	加入房间
leaveRoom	离开房间
getRoomId	获取房间id
renewAuthorization	更新签名
changeUserRole	切换角色
changeUserName	修改用户昵称

表 5-16 视频管理接口

接口	描述
setPriorRemoteVideoStreamType	设置优先选看档位
pullRemoteVideo	订阅或取消订阅远端用户
setupRemoteView	设置远端用户窗口
startRemoteStreamView	按档位选看远端用户，并设置用户窗口
stopRemoteStreamView	取消订阅远端用户，并清除此用户的窗口
pullAllRemoteVideo	订阅或取消订阅全部远端用户，对未入会用户也有影响
setRemoteVideoStreamType	切换用户指定档位的视频
updateRemoteRenderMode	设置远端视频渲染模式，镜像模式
setRemoteVideoAdjustResolution	远端流自动调整分辨率开关
startAllRemoteView	批量设置远端流视图

表 5-17 音频管理接口

接口	描述
muteRemoteAudio	远端用户静音开关
muteAllRemoteAudio	所有远端用户静音开关
adjustPlaybackVolume	调整单个用户播放音量

表 5-18 自定义渲染接口

接口	描述
setExternalAudioFrameOutputEnable	音频流第三方播放开关（音频自渲染）
setExternalVideoFrameOutputEnable	视频流第三方渲染开关（视频自渲染）
setExternalDataFrameOutputEnable	辅流第三方渲染开关（辅流自渲染）

表 5-19 辅流管理接口

接口	描述
startRemoteAuxiliaryStreamView	订阅远端用户辅流，并设置窗口
stopRemoteAuxiliaryStreamView	取消订阅远端用户辅流，并清除用户窗口
updateRemoteAuxiliaryStreamRenderMode	设置辅流渲染填充方式，镜像模式
setRemoteAuxiliaryStreamViewRotation	设置辅流旋转角度

5.4.3.2 初始化等基础接口

setEncryption

```
public abstract int setEncryption(HRTCEncryptionConfig encryptionParam);
```

【功能说明】

设置端到端加密方式。需要在加入房间前设置生效。其中sdk加密模式，需要设置16位加密密钥和加密算法，app加密模式需要先设置回调接口。

【请求参数】

encryptionParam：加密配置，具体请参见HRTCEncryptionConfig。

【返回参数】

- 0：成功。
- <0：失败。具体请参见**客户端错误码**。

release

```
public abstract void release();
```

【功能说明】

释放HRTCConnection。在leaveRoom之后调用，可释放房间占用的资源。

【请求参数】

无

【返回参数】

无

setNetworkBandwidth

```
public abstract int setNetworkBandwidth(HRTCNetworkBandwidth bandwidthParam);
```

【功能说明】

设置网络带宽限制。需要在每次加入房间之前设置。

【请求参数】

bandwidthParam：带宽设置参数，具体请参见**HRTCNetWorkBandwidth**。

【返回参数】

- 0：成功。
- <0：失败。具体请参见**客户端错误码**。

5.4.3.3 房间功能

joinRoom

```
public abstract int joinRoom(HRTCJoinParam joinParam);
```

【功能说明】

加入房间。

【请求参数】

joinParam：入会参数，具体请参见**HRTCJoinParam**。

【返回参数】

- 0：成功。
- 1：失败。
- 2：上下文为空。

【注意事项】

⚠ 注意

该方法将会触发以下回调：

- **onConnectionStateChangedNotify**：连接状态发送改变。
- **onJoinRoomSuccess**：加入房间成功时回调。
- **onJoinRoomFailure**: 加入房间失败时回调。
- **onRemoteUserOnline**：加入房间成功后，通知房间内已加入用户的回调，不包括自己。

leaveRoom

```
public abstract int leaveRoom()
```

【功能说明】

离开房间。

【请求参数】

无

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

⚠ 注意

会触发以下回调：

- **onLeaveRoom**：离开房间回调。
- **onConnectionStateChangedNotify**：连接状态改变回调。

renewAuthorization

```
public abstract int renewAuthorization(String signature, long ctime);
```

【功能说明】

鉴权签名过期，收到**onAuthorizationExpired**回调后更新鉴权签名。

【请求参数】

- signature：鉴权签名字符串。
- ctime：过期时间，单位：秒。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

getRoomId

```
public abstract String getRoomId();
```

【功能说明】

获取当前房间的roomId。

【请求参数】

无

【返回参数】

String roomId：当前房间的roomId。

changeUserRole

```
public abstract int changeUserRole(HRTCRoleType role, String authorization, long ctime);
```

【功能说明】

设置本端用户在房间内的角色。

【请求参数】

- role：用户角色，具体请参见**HRTCRoleType**。
- authorization：预留参数，填null。
- ctime：预留参数，填0。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

⚠ 注意

支持joiner，player角色间相互切换。

- 切换成功将触发**onUserRoleChangedNotify**回调。
- 切换失败将触发**onError**回调，返回错误码“HRTC_ERR_CODE_USER_ROLE_CHANGE_FAIL”。

changeUserName

```
public abstract int changeUserName(String usrName);
```

【功能说明】

修改用户昵称。

【请求参数】

usrName：用户新的昵称。昵称不为空，且最大不超过256。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

> **⚠ 注意**
>
> 修改成功后，本端会回调**onUserNameChangedNotify**事件，远端会回调**onRemoteUserNameChangedNotify**事件。

5.4.3.4 音频管理

muteRemoteAudio

```
public abstract int muteRemoteAudio(String userId, boolean mute);
```

【功能说明】

设置是否接收对应远端用户的音频流。

【请求参数】

- userId：用户ID。
- mute：true表示取消音频流接收，false表示开启音频流接收，默认为false。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

muteAllRemoteAudio

```
public abstract int muteAllRemoteAudio(boolean mute);
```

【功能说明】

设置是否接收所有远端用户的音频流。

【请求参数】

mute：true表示取消音频流接收，false表示开启音频流接收。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

> **⚠ 注意**
>
> - 取消所有音频流接收，同时也会取消接收新加入用户的音频流。
> - 开启所有音频流接收，同时也会开启接收新加入用户的音频流。
> - 默认开启所有音频流接收。

adjustPlaybackVolume

```
public abstract int adjustPlaybackVolume(String userId, int volume);
```

【功能说明】

调整单个用户播放音量增益值。

【请求参数】

- userId：用户ID。
- volume：音量值，取值范围为[0,100]，默认音量值为10无增益，10以下表示负增益，10以上表示正增益，此接口不会影响系统音量。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

5.4.3.5 视频管理

setPriorRemoteVideoStreamType

```
public abstract int setPriorRemoteVideoStreamType(HRTCVideoStreamType type);
```

【功能说明】

大小流模式，设置所有订阅的远端视频流类型。默认订阅大流，优先应用setRemoteVideoStreamType接口设置的用户流类型。

【请求参数】

type：订阅的视频流类型，分为大流和小流，具体请参见**HRTCVideoStreamType**。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

pullRemoteVideo

```
public abstract int pullRemoteVideo(String userId, boolean pull);
```

【功能说明】

开启、关闭指定远端用户的视频流。

【请求参数】

- userId：远端用户的userId，唯一标识。
- pull：true表示开始接收，false表示关闭接收。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

setupRemoteView

```
public abstract int setupRemoteView(String userId, SurfaceView view);
```

【功能说明】

设置远端流视图，该接口不影响收流。

【请求参数】

- userId：远端用户的唯一标识。
- view：远端窗口视图，view为null时，解除窗口绑定。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

startRemoteStreamView

```
public abstract int startRemoteStreamView(String userId, SurfaceView view, HRTCStreamType streamType, boolean disableAdjustRes);
```

【功能说明】

开始订阅远端视频流，并设置远端窗口。

【请求参数】

- userId：用户ID。
- view：远端窗口视图。
- streamType：流类型，具体请参见**HRTCStreamType**。
- disableAdjustRes：禁用分辨率自适应的标志。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

stopRemoteStreamView

```
public abstract int stopRemoteStreamView(String userId);
```

【功能说明】

停止订阅远端视频流，并关闭远端窗口。

【请求参数】

userId：用户ID。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

pullAllRemoteVideo

```
public abstract int pullAllRemoteVideo(boolean pull);
```

【功能说明】

批量开启、关闭当前所有远端用户的视频流。

【请求参数】

pull：true表示开启接收，false表示关闭接收， 默认开启接收。

【返回参数】

- 0：成功。
- > 0：失败。具体请参见**客户端错误码**。

setRemoteVideoStreamType

```
public abstract int setRemoteVideoStreamType(String userId, HRTCVideoStreamType type);
```

【功能说明】

大小流模式，设置指定订阅的远端视频流类型。

【请求参数】

- userId：远端用户唯一标识。
- type：订阅的视频流类型，分为大流和小流，具体请参见**HRTCVideoStreamType**。

updateRemoteRenderMode

```
public abstract int updateRemoteRenderMode(String userId, HRTCVideoDisplayMode displayMode,
HRTCVideoMirrorType mirrorMode);
```

【功能说明】

设置远端窗口渲染模式。

【请求参数】

- userId：用户ID。
- displayMode：渲染模式，具体请参见**HRTCVideoDisplayMode**。
- mirrorMode：镜像模式，具体请参见**HRTCVideoMirrorType** 。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

setRemoteVideoAdjustResolution

```
public abstract int setRemoteVideoAdjustResolution(boolean enable);
```

【功能说明】

设置是否开启远端流分辨率自适应。默认开启自适应。

【请求参数】

enable：是否开启分辨率自适应。默认开启。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

startAllRemoteView

```
public abstract int startAllRemoteView(int counts, List<HRTCVideoRemoteView> viewInfoList);
```

【功能说明】

批量设置远端流视图。

【请求参数】

- counts：必选，number类型，为数组的长度；如果设置为0，则取消所有远端流视图，大于0，则取消没选中用户的远端视图。
- viewInfoList：订阅的视图信息，主要包括该视图的句柄、流类型、用户ID、是否自适应等，具体请参见**HRTCVideoRemoteView**。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

5.4.3.6 辅流管理

startRemoteAuxiliaryStreamView

```
public abstract int startRemoteAuxiliaryStreamView(String userId, SurfaceView view);
```

【功能说明】

开始订阅辅流。

【请求参数】

- userId：用户ID。
- view：窗口视图。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

注意

- 收到**onUserAuxiliaryStreamAvailable**通知后，获取对应的userId。
- 多辅流场景下，一个用户只能同时订阅一条辅流。即当前正在订阅用户A的辅流，需要订阅另一个用户B的辅流时，需要调用startRemoteAuxiliaryStreamView停止订阅用户A的辅流后，才能订阅用户B的辅流。

stopRemoteAuxiliaryStreamView

```
public abstract int stopRemoteAuxiliaryStreamView(String userId);
```

【功能说明】

停止订阅辅流。

【请求参数】

userId：用户ID。

【返回参数】

- 0：成功。

- >0：失败。具体请参见**客户端错误码**。

setRemoteAuxiliaryStreamViewRotation

```
public abstract int setRemoteAuxiliaryStreamViewRotation(String userId, HRTCRotationType rotation);
```

【功能说明】

设置辅流角度。

【请求参数】

- userId：用户ID。
- rotation：辅流角度，默认值为HRTC_ROTATION_TYPE_0，具体请参见**HRTCRotationType**。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

updateRemoteAuxiliaryStreamRenderMode

```
public abstract int updateRemoteAuxiliaryStreamRenderMode(String userId, HRTCVideoDisplayMode displayMode, HRTCVideoMirrorType mirrorMode);
```

【功能说明】

设置辅流渲染模式。

【请求参数】

- userId：用户ID。
- displayMode：渲染模式，默认值为HRTC_VIDEO_DISPLAY_MODE_FIT，具体请参见**HRTCVideoDisplayMode**。
- mirrorMode：镜像模式，默认值为HRTC_VIDEO_MIRROR_TYPE_DISABLE，具体请参见**HRTCVideoMirrorType**。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

5.4.3.7 自定义渲染

setExternalAudioFrameOutputEnable

```
public abstract int setExternalAudioFrameOutputEnable(boolean localEnable,boolean remoteEnable);
```

【功能说明】

设置音频数据输出使能。

【请求参数】

- localEnable：true表示输出本地音频数据，false表示不输出本地音频数据。
- remoteEnable：true表示输出远端音频数据，false表示不输出远端音频数据。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

⚠ 注意

通过**onPlaybackExternalAudioFrame**回调音频数据。

setExternalVideoFrameOutputEnable

```
public abstract int setExternalVideoFrameOutputEnable(boolean localEnable,boolean remoteEnable);
public abstract int setExternalVideoFrameOutputEnable(boolean localEnable, boolean remoteEnable,
HRTCImageBufferFormat format);
```

【功能说明】

设置视频数据输出使能。

【请求参数】

- localEnable：true表示输出本地视频数据，false表示不输出本地视频数据，默认值为false。
- remoteEnable：true表示输出远端视频数据，false表示不输出远端视频数据，默认值为false。
- format：自渲染输出的视频帧图片格式，在**onRenderExternalVideoFrame**接口参数的videoFrameType中体现，取值请参见**HRTCImageBufferFormat**。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

setAuxiliaryExternalVideoFrameOutputEnable

```
public abstract int setAuxiliaryExternalVideoFrameOutputEnable(boolean localEnable,boolean
remoteEnable);
```

【功能说明】

设置共享辅流数据输出使能。Android平台接口暂不提供本地共享数据的设置。

【请求参数】

- localEnable：true表示输出本地共享数据，false表示不输出本地共享数据，默认值为false。
- remoteEnable：true表示输出远端共享数据，false表示不输出远端共享数据，默认值为false。

【返回参数】

- 0：成功。
- >0：失败。具体请参见**客户端错误码**。

5.4.4 事件回调(IHRTCConnectionEventHandler)

本章节介绍了Android SDK的回调接口IHRTCConnectionEventHandler的详情。

表 5-20 回调接口

接口	描述
onError	错误回调
onJoinRoomSuccess	加入房间成功回调
onJoinRoomFailure	加入房间失败回调
onRejoinRoomSuccess	重新加入房间回调
onLeaveRoom	离开房间回调
onRemoteUserOnline	用户加入房间回调
onRemoteUserOffline	用户离开房间回调
onConnectionChangedNotify	网络连接状态改变回调
onRemoteVideoStateChangedNotify	远端视频流状态变化回调
onRemoteAudioStateChangedNotify	远端音频流状态变化回调
onUserRoleChangedNotify	用户角色改变回调
onUserVolumeStatsNotify	提示频道远端用户以及自己的音量回调
onUserAuxiliaryStreamAvailable	辅流加入房间回调
onVideoStatsNotify	视频流详情，2s触发一次回调
onAudioStatsNotify	音频流详情，2s触发一次回调
onAuxiliaryStreamStatsNotify	辅流详情，2s触发一次回调
onFirstRemoteVideoDecoded	远端用户第一帧解码成功通知
onFirstRemoteAuxiliaryStreamDecoded	远端用户辅流第一帧解码成功通知
onRemoteUserNameChangedNotify	远端用户昵称变更通知
onUserNameChangedNotify	本端修改昵称结果通知
onAuthorizationExpired	签名过期回调
onMediaConnectStateChangedNotify	媒体连接状态变化通知
onRenderSuccessNotify	用户视频流渲染成功通知回调
onMediaStreamRecvPktNotify	订阅的用户视频流收包信息回调
onNetworkQualityNotify	加入房间后的网络质量状态回调
onRemoteVideoStatsNotify	远端视频统计回调

接口	描述
onRemoteAudioStatsNotify	远端音频统计回调
onVideoResolutionChangedNotify	远端视频大小流改变回调
onStatsNotify	当前会话统计回调
onStartPublishStream	开始旁路（RTMP）推流回调
onUpdateTransCoding	更新旁路（RTMP）推流消息
onStopPublishStream	停止旁路（RTMP）推流消息
onStreamPublishStateChange	RTMP推流状态回调
onTopActiveSpeaker	远端音频最活跃用户回调
onRemoteMicrophoneStateChanged	远端麦克风设备状态变更通知

onError

```
void onError(HRTCConnection conn, int error, String msg);
```

【功能说明】

错误回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- error：错误码，具体请参见HRTCErrorCode。
- msg：错误描述。

onJoinRoomSuccess

```
void onJoinRoomSuccess(HRTCConnection conn, String userId);
```

【功能说明】

成功加入房间回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- roomId：房间ID。
- userId：用户ID。

onJoinRoomFailure

```
void onJoinRoomFailure(HRTCConnection conn, int error, String msg);
```

【功能说明】

加入房间失败回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- error：错误码。
- msg：错误信息。

onLeaveRoom

```
void onLeaveRoom(HRTCConnection conn, HRTCLeaveReason reason, HRTCStatsInfo statsInfo);
```

【功能说明】

离开房间回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- reason：自己离开房间的原因，具体请参见**HRTCLeaveReason**。
- statsInfo：会议期间的统计信息，具体请参见**HRTCStatsInfo**。

onRemoteUserOnline

```
void onRemoteUserOnline(HRTCConnection conn, String userId, String userName);
```

【功能说明】

用户加入房间成功回调，不包括自己。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- userId：用户ID。
- userName：用户名称。

onRemoteUserOffline

```
void onRemoteUserOffline(HRTCConnection conn, String userId, int reason);
```

【功能说明】

用户离开房间回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- userId：用户ID。
- reason：预留字段。

onConnectionChangedNotify

```
void onConnectionChangedNotify(HRTCConnection conn, HRTCConnStateTypes connStateTypes,
HRTCConnChangeReason connChangeReason, String description);
```

【功能说明】

网络连接状态改变。

【回调参数】

- conn：回调对应的HRTCConnection实例。

- connStateTypes：网络连接状态，具体请参见**HRTCConnStateTypes**。
- connChangeReason：网络连接状态原因，具体请参见**HRTCConnChangeReason**。
- description：描述。

onUserRoleChangedNotify

```
void onUserRoleChangedNotify(HRTCConnection conn, HRTCRoleType oldRole, HRTCRoleType newRole);
```

【功能说明】

用户角色改变。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- oldRole：改变前的角色，具体请参见**HRTCRoleType**。
- newRole：改变后的角色，具体请参见**HRTCRoleType**。

onAudioRouteStateChangedNotify

```
void onAudioRouteStateChangedNotify(HRTCConnection conn, HRTCAudioRoute route);
```

【功能说明】

音频路由改变。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- route：音频路由，具体请参见**HRTCAudioRoute**。

onUserAuxiliaryStreamAvailable

```
void onUserAuxiliaryStreamAvailable(HRTCConnection conn, String userId, boolean available);
```

【功能说明】

辅流加入房间。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- userId：用户ID。
- available：true为辅流开始推送，false为辅流停止推送。

onVideoStatsNotify

```
void onVideoStatsNotify(HRTCConnection conn, List<HRTCLocalVideoStats> localStats,
List<HRTCRemoteVideoStats> remoteStats);
```

【功能说明】

视频流详情，2s触发一次回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。

- localStats：本地视频发流详情参数，具体请参见**HRTCLocalVideoStats**。
- remoteStats：远端视频收流详情参数，具体请参见**HRTCRemoteVideoStats**。

onAudioStatsNotify

```
void onAudioStatsNotify(HRTCConnection conn, List<HRTCLocalAudioStats> localStats,
List<HRTCRemoteAudioStats> remoteStats);
```

【功能说明】

音频流详情，2s触发一次回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- localStats：本地音频发流详情，具体请参见**HRTCLocalAudioStats**。
- remoteStats：远端音频收流详情，具体请参见**HRTCRemoteAudioStats**。

onAuxiliaryStreamStatsNotify

```
void onAuxiliaryStreamStatsNotify(HRTCConnection conn, List<HRTCLocalVideoStats> localStats,
List<HRTCRemoteVideoStats> remoteStats);
```

【功能说明】

辅流详情，2s触发一次回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- localStats：本地辅流的发流详情，具体请参见**HRTCLocalVideoStats**。
- remoteStats：远端辅流的收流详情，具体请参见**HRTCRemoteVideoStats**。

onRemoteAudioStateChangedNotify

```
void onRemoteAudioStateChangedNotify(HRTCConnection conn, String userId,
HRTCRemoteAudioStreamState state, HRTCRemoteAudioStreamStateReason reason);
```

【功能说明】

远端音频流状态变化回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- userId：远端用户ID。
- state：远端音频流状态，具体请参见**HRTCRemoteAudioStreamState**。
- reason：远端音频流状态变化原因，具体请参见**HRTCRemoteAudioStreamStateReason**。

onRemoteVideoStateChangedNotify

```
void onRemoteVideoStateChangedNotify(HRTCConnection conn, String userId,
HRTCRemoteVideoStreamState state, HRTCRemoteVideoStreamStateReason reason);
```

【功能说明】

远端视频流状态变化回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- userId：远端用户ID。
- state：远端视频流状态，具体请参见**HRTCRemoteVideoStreamState**。
- reason：远端视频流状态变化原因，具体请参见**HRTCRemoteVideoStreamStateReason**。

onRejoinRoomSuccess

```
void onRejoinRoomSuccess(HRTCConnection conn, String userId);
```

【功能说明】

重新加入房间回调。例如网络异常后重连成功加入房间时触发。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- userId：用户ID。

onUserVolumeStatsNotify

```
void onUserVolumeStatsNotify(HRTCConnection conn, List<HRTCVolumeInfo> volumeInfos, int
totalVolume);
```

【功能说明】

提示频道远端用户以及自己的音量回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- volumeInfos：回调发言人信息列表，目前最多支持4人，包括自己，具体请参见**HRTCVolumeInfo**。
- totalVolume：远端混音后的总音量。

onFirstRemoteVideoDecoded

```
void onFirstRemoteVideoDecoded(HRTCConnection conn, String userId, int width, int height);
```

【功能说明】

远端用户第一帧解码成功回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- userId：用户ID。
- width：视频宽度。
- height：视频高度。

onFirstRemoteAuxiliaryStreamDecoded

```
void onFirstRemoteAuxiliaryStreamDecoded(HRTCConnection conn, String roomId, String userId, int width,
int height);
```

【功能说明】

接收到第一帧远端辅流并解码成功，触发此回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- roomId：视频流对应的房间ID。
- userId：视频流对应的用户ID。
- width：视频流宽。
- height：视频流高。

onAuthorizationExpired

```
void onAuthorizationExpired(HRTCConnection conn);
```

【功能说明】

签名过期回调，需要app调用renewAuthorization更新签名。

【回调参数】

conn：回调对应的HRTCConnection实例。

onRenderSuccessNotify

```
void onRenderSuccessNotify(HRTCConnection conn, String userId, boolean isAux);
```

【功能说明】

用户视频流渲染成功通知回调。首帧渲染成功、分辨率变化或视频流中断后恢复触发。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- userId：用户ID。
- isAux：是否辅流。true为是辅流，false为不是辅流。

onMediaStreamRecvPktNotify

```
void onMediaStreamRecvPktNotify(HRTCConnection conn, List<HRTCStreamPacketInfo>
streamPacketInfoList);
```

【功能说明】

订阅的用户视频流收包信息回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- streamPacketInfoList：订阅的用户视频流收包信息，具体请参见**HRTCStreamPacketInfo**。

onNetworkQualityNotify

```
void onNetworkQualityNotify(HRTCConnection conn, List<HRTCQualityInfo> upStreamQuality,
List<HRTCQualityInfo> downStreamQuality);
```

【功能说明】

加入房间后，基于流级别的网络质量状态回调，音频流、视频流分开回调。

【回调参数】

- 当远端订阅本端时（本端开始推流），本端才会开始收到网络质量回调。
- conn：回调对应的HRTCConnection实例。
- upStreamQuality：上行网络质量状态，具体请参见**HRTCQualityInfo**。当前不可用。
- downStreamQuality：下行网络质量状态，具体请参见**HRTCQualityInfo**。

onMediaConnectStateChangedNotify

```
void onMediaConnectStateChangedNotify(HRTCConnection conn, HRTCMediaConnStateTypes state,
HRTCMediaConnChangeReason reason, String description);
```

【功能说明】

媒体服务器连接状态变更通知。加入房间过后，收到媒体服务的数据包时，返回Connected消息，超过6s没有收到包，则返回Failed消息。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- state：连接状态，具体请参见**HRTCMediaConnStateTypes**。
- reason：连接状态变化的原因，具体请参见**HRTCMediaConnChangeReason**。
- description：连接状态变化原因描述。

onUserNameChangedNotify

```
void onUserNameChangedNotify(HRTCConnection conn, String oldUserName, String newUserName);
```

【功能说明】

调用changeUserName接口成功后，上报此事件通知修改昵称结果。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- oldUserName：修改前的昵称。
- newUserName：修改后的昵称。

onRemoteUserNameChangedNotify

```
void onRemoteUserNameChangedNotify(HRTCConnection conn, String userId, String userName);
```

【功能说明】

远端用户修改昵称后，通知本端昵称变更。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- userId：修改昵称的用户id。
- userName：修改后的昵称。

onRemoteVideoStatsNotify

```
void onRemoteVideoStatsNotify(HRTCConnection conn, List<HRTCRemoteVideoStats> remoteStats);
```

【功能说明】

远端视频流详情，2s触发一次回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- remoteStats：远端视频收流详情参数，具体请参见**HRTCRemoteVideoStats**。

onRemoteAudioStatsNotify

```
void onRemoteAudioStatsNotify(HRTCConnection conn, List<HRTCRemoteAudioStats> remoteStats);
```

【功能说明】

远端音频流详情，2s触发一次回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- remoteStats：远端音频收流详情，具体请参见**HRTCRemoteAudioStats**。

onVideoResolutionChangedNotify

```
void onVideoResolutionChangedNotify(HRTCConnection conn, String userid, int width, int height);
```

【功能说明】

远端视频大小改变回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- userId：用户ID。
- width：视频流宽度。
- height：视频流高度。

onStatsNotify

```
void onStatsNotify(HRTCConnection conn,List<HRTCOnStats> hrtcOnStats);
```

【功能说明】

当前会话统计回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- hrtcOnStats：会话统计信息，具体请参见**HRTCOnStats**。

onStartPublishStream

```
void onStartPublishStream(HRTCConnection conn,int code, const char* taskId);
```

【功能说明】

开始旁路（RTMP）推流回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- code：错误码，0表示成功，非0请参见**错误码**。
- taskId：任务Id。

onUpdateTransCoding

```
void onUpdateTransCoding(HRTCConnection conn,int code, const char* taskId);
```

【功能说明】

更新旁路（RTMP）推流回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- code：错误码，0表示成功，非0请参见**错误码**。
- taskId：任务Id。

onStopPublishStream

```
void onStopPublishStream(HRTCConnection conn,int code, const char* taskId);
```

【功能说明】

停止旁路（RTMP）推流回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- code：错误码，0表示成功，非0请参见**错误码**。
- taskId：任务Id。

onStreamPublishStateChange

```
void onStreamPublishStateChange(HRTCConnection conn,int code, const char* taskId, const
HRTCUrlStatusList * urlStatus);
```

【功能说明】

RTMP推流状态回调。

【回调参数】

- conn：回调对应的HRTCConnection实例。
- code：错误码，0表示成功，非0请参见**错误码**。
- taskId：任务Id。
- urlStatus：推流的url结果结构体，具体请参见**HRTCUrlStatusList**。

onTopActiveSpeaker

```
public void onTopActiveSpeaker(HRTCConnection conn, String userId, boolean noStream);
```

【功能说明】

上报当前最活跃的用户userId。该回调主要用于0号会场场景（额外订阅uid为0的用户音频）。

【 回调参数 】

userId：返回当前远端音量最活跃的用户ID。

noStream：该用户是否有视频流。

⚠ 注意

0号会场模式下，SDK会持续监测（根据一定时间内用户音量大小）当前最活跃的用户，如果最活跃用户发生变化，则触发此回调并上报当前最活跃的用户userId。

onRemoteMicrophoneStateChanged

```
void onRemoteMicrophoneStateChanged(HRTCConnection conn, String userId, HRTCRemoteMicState state);
```

【 功能说明 】

远端麦克风设备状态变更回调。

【 回调参数 】

- conn: 连接对象。
- userId：用户id。
- state：麦克风设备状态，具体请参见**HRTCRemoteMicState**。

onRemoteShareStatusChangeNotify

```
void onRemoteShareStatusChangeNotify(HRTCConnection conn, String userId, boolean isRecving)
```

【 功能说明 】

上报下行辅流状态变化通知，接收端探测网络变化，分配给下行辅流，如果有辅流因为带宽不足被停掉或者带宽恢复而恢复，则上报给产品。

【 回调参数 】

conn: 连接对象。

userId：下行辅流对应的用户ID。

isRecving：是否接受。

5.4.5 客户端错误码

本章节介绍了SDK的客户端错误码HRtcErrorCode的详细信息。

表 5-21 类成员说明

类成员	错误码	描述	错误原因
HRTC_ERR_CODE_SUCCESS	0	成功	-
HRTC_ERR_CODE_SDK_INTERNAL_ERROR	90000001	SDK内部系统错误	SDK内部异常。

类成员	错误码	描述	错误原因
HRTC_ERR_CODE_MSG_TOOLARGE	90000002	发送的消息太大	发送消息时，消息体太大。
HRTC_ERR_CODE_MEM_NOT_ENOUGH	90000003	内存不足	内存申请不到。
HRTC_ERR_CODE_SYNSEND_MSG_ERR	90000004	消息发送失败	消息队列异常，导致内部消息发送失败。
HRTC_ERR_CODE_PARAM_ERROR	90000005	参数错误	包括如下两方面： • 接口入参无效； • 内部参数错误。
HRTC_ERR_CODE_API_CALLED_IN_WRONG_ORDER	90000006	API调用顺序不当	当前只有日志设置必须在初始化之前。
HRTC_ERR_CODE_SETUP_LOCAL_VIEW_FAIL	90000007	设置本地窗口失败	内部窗口占用太多，无法再设置本地窗口。
HRTC_ERR_CODE_START_REMOTE_STREAM_VIEW_FAIL	90000008	设置远端窗口失败	publisher场景没有远端画面，不应该设置。或者内部窗口占用太多，无法再设置远端窗口。
HRTC_ERR_CODE_SET_DEVICE_FAIL	90000009	设置设备失败	设置播放、录音、视频设备失败。
HRTC_ERR_CODE_INITIALIZING	90000010	初始化过程中	初始化过程中，不能再做初始化或者去初始化操作。
HRTC_ERR_CODE_UN_INITIALIZING	90000011	去初始化过程中	去初始化过程中，不能再做初始化或者去初始化操作。
HRTC_ERR_CODE_LOG_UPLOADING	90000012	日志正在上传	日志正在上传过程中。
HRTC_ERR_CODE_MEDIA_PORT_ERROR	90000013	媒体端口获取失败	音频从10010开始，视频从10020开始，尝试10次，端口都被占用了。
HRTC_ERR_CODE_WATCH_VIEW_TOO_MUCH	90000014	选看视频超过规格	当前规格为16，超过16个设置远端窗口，就会失败。
HRTC_ERR_CODE_MEDIA_CMP_ERR	90000015	媒体协商失败	与服务器之间媒体协商失败。
HRTC_ERR_CODE_SERVER_NO_RESPONSE	90000016	服务器没有响应	加入房间，选看在2-4s内没有收到服务器的响应。

类成员	错误码	描述	错误原因
HRTC_ERR_CODE_USER_ROLE_CHANGE_FAIL	90000017	角色切换失败	角色切换失败。
HRTC_ERR_CODE_JOIN_ROOM_FAIL	90000018	加入房间失败	加入房间失败。
HRTC_ERR_CODE_JOIN_ROOM_STATUS_BUSY	90000019	加入房间失败	用户已经在房间里，或正在进行网络探测。
HRTC_ERR_CODE_JOIN_ROOM_SERVER_ERROR	90000020	加入房间失败	加入房间失败，服务器异常。
HRTC_ERR_CODE_JOIN_ROOM_SERVICE_UNREACHABLE	90000021	加入房间失败	加入房间失败，服务不可达。
HRTC_ERR_CODE_JOIN_ROOM_AUTH_FAIL	90000022	加入房间失败	加入房间失败，鉴权失败。
HRTC_ERR_CODE_JOIN_ROOM_AUTH_RETRY	90000023	加入房间失败	加入房间失败，鉴权重试。
HRTC_ERR_CODE_JOIN_ROOM_AUTH_CLOCK_SYNC	90000024	加入房间失败	加入房间失败，时钟同步。
HRTC_ERR_CODE_JOIN_ROOM_URL_NOT_RIGHT	90000025	加入房间失败	加入房间失败，url错误。
HRTC_ERR_CODE_KICKED_OFF	90000026	被踢出房间	相同用户ID等原因，被踢出房间。
HRTC_ERR_CODE_SCREEN_CAPTURE_FAIL	90000027	屏幕共享失败	屏幕共享失败。
HRTC_ERR_CODE_EXT_MEDIA_OUTPUT	90000028	设置输出设备错误	当开启媒体数据输出时，不允许设置输出设备，否则会报此异常。
HRTC_ERR_CODE_RECONNECT_FAILED	90000029	重连失败	重连失败。
HRTC_ERR_CODE_SERVER_BREAK_DOWN	90000030	服务器断开	服务器断开。
HRTC_ERR_CODE_SIGNATURE_EXPIRED	90000031	签名过期	签名过期。

类成员	错误码	描述	错误原因
HRTC_ERR_CODE_SET_REMOTE_RENDER_MODE_FAIL	90000032	设置远端渲染模式失败	设置远端渲染模式失败。
HRTC_ERR_CODE_SET_REMOTE_AUDIO_MUTE_FAIL	90000033	设置远端音频静音失败	当前的模式不支持设置远端静音，或者设置的远端静音状态不正确。
HRTC_ERR_CODE_SET_USEROLE_NOT_ALLOWED	90000036	不允许角色切换	跨房时本房间内角色切换失败，通过onConnectOtherRoom返回。
HRTC_ERR_CODE_EXT_MEDIA_CAPTURE_INPUT	90000037	当前为第三方采集模式，禁用该操作	处于第三方采集模式下，修改视频输入设备时，上报此错误码。
HRTC_ERR_CODE_SET_EXTAUDIO_CAPTURE_FAIL	90000038	设置第三方音频采集失败	已经加入房间后，调用**setExternalAudioCapture**接口，会上报此错误码。
HRTC_ERR_CODE_SET_EXTVIDEO_CAPTURE_FAIL	90000039	设置第三方视频采集失败	已经加入房间后，调用**setExternalVideoCapture**接口，会上报此错误码。
HRTC_ERR_CODE_SET_SHARE_COMPUTER_SOUND_FAIL	90000040	设置共享声音开关失败	设置共享声音开关失败。
HRTC_ERR_CODE_SET_LOCAL_AUDIO_MUTE_FAIL	90000041	启停上行音频流失败	启停上行音频流失败。
HRTC_ERR_CODE_SET_LOCAL_VIDEO_MUTE_FAIL	90000042	启停上行视频流失败	启停上行视频流失败。
HRTC_ERR_CODE_USER_REMOVED	90000043	用户被移除	用户被移除。
HRTC_ERR_CODE_ROOM_DISMISSED	90000044	房间被解散	房间被解散。
HRTC_ERR_CODE_SETUP_REMOTE_VIEW_FAIL	90000045	设置远端View失败	设置远端View失败。

类成员	错误码	描述	错误原因
HRTC_ERR_CODE_LOCAL_AUDIO_DISABLE_FAIL	90000056	当前未推音频流	当前未推音频流。
HRTC_ERR_COD_ROLE_NOT_SUPPORT	90000057	当前用户角色不支持	当前用户角色不支持该操作。
HRTC_ERR_CODE_NOT_INCLUDE_MLSDK	90000058	没有动态加载ML图像分割库，不能支持背景虚化和背景替换能力	没有动态加载ML图像分割库，不能支持背景虚化和背景替换能力。
HRTC_ERR_CODE_ENABLE_BACKGROUND_FAIL	90000059	背景虚化或背景替换开启失败	背景虚化或背景替换开启失败。

5.4.6 服务端错误码

当SDK运行出现网络、媒体相关等错误时，SDK无法自动恢复，需要APP干预或进行用户提示。该错误码由服务端产生，通过onError返回。

表 5-22 服务端错误码

错误码	描述	错误原因
RTC.10000001	内部错误	程序或环境问题
RTC.31000000	节点不存在	程序或环境问题
RTC.31000001	session校验失败	程序或环境问题
RTC.31000003	内部异常	程序或环境问题
RTC.31000004	认证失败	用户使用问题
RTC.31000005	请重试	用户使用问题
RTC.31000006	需要时钟同步	用户使用问题
RTC.31000007	请求资源不存在	程序或环境问题
RTC.32000000	服务异常	程序或环境问题

错误码	描述	错误原因
RTC.32000001	流号已满	程序或环境问题
RTC.32000002	SFU为空	程序或环境问题
RTC.32000004	下发流信息失败	程序或环境问题
RTC.32000005	添加适配器失败	程序或环境问题
RTC.32000006	添加路由失败	程序或环境问题
RTC.32000007	获取用户信息异常	程序或环境问题
RTC.32000010	选看用户不存在	程序或环境问题
RTC.32000011	音频速率参数非法	程序或环境问题
RTC.32000012	用户列表为空	程序或环境问题
RTC.32000013	非法请求参数	程序或环境问题
RTC.32000015	内部调用异常	程序或环境问题
RTC.32000016	内部调用异常	程序或环境问题
RTC.32000017	站点不存在	程序或环境问题
RTC.32000018	错误的加密算法	程序或环境问题
RTC.32000019	客户端媒体加密密钥base64解码失败	程序或环境问题
RTC.32000020	生成媒体加密密钥失败	程序或环境问题
RTC.32000021	下发加密信息异常	程序或环境问题
RTC.32000022	获取SFU的IP失败	程序或环境问题
RTC.32000024	内部调用异常	程序或环境问题
RTC.32000025	内部调用异常	程序或环境问题
RTC.32000028	不支持的操作	程序或环境问题
RTC.32000030	sfu资源不足	程序或环境问题
RTC.32000032	跨房数量超过上限	用户使用问题
RTC.32000033	不允许重复跨入同一房间	用户使用问题
RTC.32000034	稍后重试	程序或环境问题
RTC.33000000	服务异常	程序或环境问题
RTC.33000001	节点不存在	程序或环境问题
RTC.34000001	房间已满	用户使用问题
RTC.34000002	房间不存在	程序或环境问题

错误码	描述	错误原因
RTC.34000003	站点不存在	程序或环境问题
RTC.34000004	内部调用异常	程序或环境问题
RTC.34000006	用户不存在	用户使用问题
RTC.34000007	已存在辅流共享	用户使用问题

5.4.7 数据类型

本章节列出了Android SDK的所有数据类型，您可以结合HRtcEngine接口和回调进行开发。

表 5-23 数据类型

类型	描述
HRTCLogInfo	日志信息
HRTCCameraConfig	相机配置
HRTCEncryptionConfig	端到端加密参数
HRTCUserInfo	用户信息
HRTCJoinParam	入会参数
HRTCStatsInfo	卡顿统计信息
HRTCVideoEncParam	视频编码参数列表
HRTCLocalVideoStats	本地视频流信息
HRTCLocalAudioStats	本地音频流信息
HRTCRemoteVideoStats	远端视频流信息
HRTCRemoteAudioStats	远端音频流信息
HRTCConnectInfo	跨房信息
HRTCVideoFrame	视频帧信息
HRTCAudioFrame	音频帧信息
HRTCFrameBuffer	媒体帧信息
HRTCLogLevel	日志级别
HRTCMediaType	媒体类型
HRTCSpeakerModel	声音播放模式
HRTCAudioRoute	音频路由

类型	描述
HRTCStreamType	流类型
HRTCRoleType	角色类型
HRTCVideoDisplayMode	图像填充模式
HRTCConnStateTypes	网络连接状态
HRTCConnChangeReason	网络连接状态发生变化原因
HRTCRotationType	辅流角度
HRTCVideoFrameFormat	视频格式
HRTCAudioFrameType	音频格式
HRTCLeaveReason	离开房间的原因
HRTCVideoMirrorType	镜像模式类型
HRTCRemoteAudioStreamStateReason	远端音频流状态发生变化原因
HRTCRemoteAudioStreamState	远端音频流状态
HRTCRemoteVideoStreamState	远端视频流状态
HRTCRemoteVideoStreamStateReason	远端视频流状态发生变化原因
HRTCVolumeInfo	发言人音量
HRTCNetworkTestConfig	网络探测参数
HRTCNetworkTestResult	网络探测结果数据
HRTCNetworkQualityLevel	网络质量级别
HRTCNetworkTestState	网络探测结果状态
HRTCNetworkTestResultParam	网络探测结果参数
HRTCMediaDirection	媒体方向指示
HRTCVideoStreamType	视频流类型
HWRtcVideoEncodeResolutionMode	视频编码比例模式
HRTCQualityInfo	网络质量信息
HRTCLocalAudioStreamStateReason	本地音频流状态发生变化原因
HRTCLocalAudioStreamState	本地音频流状态
HRTCLocalVideoStreamState	本地视频流状态
HRTCLocalVideoStreamStateReason	本地视频流状态发生变化原因
HRTCRemoteAudioMode	远端音频模式

类型	描述
HRTCMediaConnStateTypes	媒体服务器状态变更类型
HRTCMediaConnChangeReason	媒体连接状态变更原因
HRTCVideoAuxiliaryEncParam	辅流的编码参数
HRTCVideoImageBufferType	视频原始数据帧内图像数据类型
HRTCImageBufferFormat	视频原始数据格式类型
HRTCEngineConfig	引擎创建相关参数
HRTCAreaCode	服务器的访问区域
HRTCOnStats	会话统计信息
HRTCAudioFileState	音频文件播放状态
HRTCAudioFileReason	音频文件播放状态变化原因
HRTCMediaOptions	媒体选参
HRTCEngineContext	引擎初始化参数
HRTCLogConfig	日志配置参数
HRTCSfuType	SFU类型
HRTCDeviceType	系统音视频设备设备类型
HRTCDeviceState	系统音视频设备设备状态
HRTCUrlStatusList	rtmp推流回调url状态列表
HRTCVideoRemoteView	远端流视图
HRTCRemoteMicState	远端麦克风设备状态
HRTCMultiRoomMediaRelayConfiguration	跨房配置
HRTCSrcMultiRoomMediaInfo	源房间信息
HRTCDstMultiRoomMediaInfo	目标房间信息
HRTCMultiRoomMediaRelayState	跨房状态
HRTCMultiRoomMediaRelayStateCode	跨房状态码

HRTCLogInfo

表 5-24 日志信息

属性	类型	描述
level	HRTCLogLevel	日志级别
path	String	日志存储路径

HRTCCameraConfig

表 5-25 相机配置

属性	类型	描述
cameraDirection	HRTCCameraDirection	相机方向，只适合移动端选择（Android和iOS）

HRTCCameraDirection

表 5-26 相机方向

枚举值	描述
HRTC_CAMERA_REAR	后置摄像头
HRTC_CAMERA_FRONT	前置摄像头

HRTCEncryptionConfig

表 5-27 端到端加密参数

属性	类型	描述
cryptionMode	HRTCCryptionMode	加密模式
cryptionSec	String	加密密钥，仅sdk加密模式下需要设置。必须是长度大于等于32的16进制字符串。
suiteType	HRTCSuiteType	加密算法，当前仅支持HRTC_ENCRYPTION_128_CTR，sdk加密模式下需要设置
secFormat	HRTCCryptionSecFormat	密钥格式，当前只支持16进制字符串。

HRTCCryptionSecFormat

表 5-28 密钥格式。

枚举值	描述
HRTC_HEX_STRING	默认模式，16进制字符串。当前只支持此格式。

HRTCCryptionMode

表 5-29 端到端加密模式

枚举值	描述
HRTC_CRYPTO_DEFAULT	默认模式，srtp认证+加密
HRTC_CRYPTO_AUTHENTICATION_SDK	sdk加密，srtp认证
HRTC_CRYPTO_AUTHENTICATION_APP	app层加密，srtp认证

HRTCUserInfo

表 5-30 用户信息

属性	类型	描述
userId	String	用户ID
userName	String	用户名，UTF-8编码格式，可选。
ctime	long	签名时间戳，单位秒，有signature时必选。
signature	String	签名，可选，具体生成方法请参见**接入鉴权**。
role	**HRTCRoleType**	角色
optionalInfo	String	可选JSON字串[{key:param1,value:value1},{key:param2,value:value2}]

HRTCJoinParam

表 5-31 入会参数

属性	类型	描述
userId	String	用户ID，支持最大长度64，支持数字、字母大小写、下划线、中线、"."字符。
userName	String	用户名，UTF-8编码格式，可选，支持最大长度128。
ctime	long	签名时间戳，单位秒，有signature时必选。
authorization	String	签名，必填，鉴权私钥请在**应用管理**中获取。签名的具体生成方法请参见**接入鉴权**。支持最大长度为1024。
role	**HRTCRoleType**	角色。
optionalInfo	String	可选JSON字串[{key:param1,value:value1},{key:param2,value:value2}]
roomId	String	房间Id，支持最大长度64，支持数字、字母大小写、下划线、中线字符。
autoSubscribeVideo	boolean	是否自动订阅视频。
autoSubscribeAudio	boolean	是否自动订阅音频。
scenario	**HRTCRemoteAudioMode**	使用的场景。 • 0：主动订阅（默认）。 • 1：TopN（千人）。 • 2：P2P。 • 3：RTSA CMD自动订阅。

HRTCDeviceType

表 5-32 系统音视频设备类型

枚举值	描述
HRTC_DEVTYPE_AUDIO_PLAYBACK	音频播放设备。
HRTC_DEVTYPE_AUDIO_RECORDING	音频录制设备。

枚举值	描述
HRTC_DEVTYPE_VIDEO_CAPTURE	视频采集设备。

HRTCStatsInfo

表 5-33 统计信息

属性	类型	描述
mildlyFrozenCounts	long	600ms卡顿次数
severelyFrozenCounts	long	超过1s卡顿次数
totalMildlyFrozenTime	long	600ms卡顿总时长
totalSeverelyFrozenTime	long	1s卡顿总时长
totalActiveTime	long	总时间，包括每一路选看的视频流启动到停止的时间总和

HRTCVideoEncParam

表 5-34 视频编码参数属性说明

属性	类型	描述
streamType	**HRTCStreamType**	视频流类型，根据**HRTCStreamType**和**表14 不同分辨率下帧率和码率的推荐值**设置需要的分辨率和宽高比
width	int	视频宽度
height	int	视频高度
frameRate	int	视频帧率，可参考**表5-36**中的帧率进行设置
minFrameRate	int	视频最小帧率，大于等于0，小于等于frameRate
bitRate	int	视频码率，可参考**表5-36**中的码率进行设置
minBitrate	int	视频最小码率，大于等于0，小于等于bitrate

表 5-35 视频编码参数方法说明

方法	描述
public HRTCVideoEncParam(HRTCStreamType streamType, int width, int height, int frameRate, int minFrameRate, int bitrate, int minBitrate);	【功能说明】 HRTCVideoEncParam构造函数。 【请求参数】 • streamType：视频流类型，具体请参见**HRTCStreamType**。 • width：视频宽度。 • height：视频高度。 • frameRate：视频帧率。 • minFrameRate：视频最小帧率。 • bitrate：视频码率。 • minBitrate：视频最小码率。 【返回参数】 HRTCVideoEncParam类对象。
public HRTCStreamType getStreamType();	【功能说明】 获取视频流类型。 【请求参数】 无 【返回参数】 视频流类型，具体参见**HRTCStreamType**。
public void setStreamType(HRTCStreamType streamType);	【功能说明】 设置视频流类型。 【请求参数】 streamType：流类型，具体请参见**HRTCStreamType**。 【返回参数】 视频流类型，具体参见**HRTCStreamType**。

表 5-36 不同分辨率下帧率和码率的推荐值

分辨率	分辨率类型	比例	最小帧率（fps）	最大帧率（fps）	最小码率	最大码率
160 X 90	LD	16:9	10	30	64	270
320 X 180	SD	16:9	10	30	80	600
480 X 270	HD	16:9	10	30	160	1050
640 X 360	HD	16:9	10	30	200	1700

分辨率	分辨率类型	比例	最小帧率（fps）	最大帧率（fps）	最小码率	最大码率
800 X 450	FHD	16:9	10	30	300	2100
960 X 540	FHD	16:9	10	30	400	2400
1120 X 630	FHD	16:9	10	30	450	2800
1280 X 720	FHD	16:9	10	30	500	4000
120 X 90	LD	4:3	10	30	64	240
160 X 120	SD	4:3	10	30	64	270
240 X 180	SD	4:3	10	30	80	450
320 X 240	HD	4:3	10	30	100	600
400 X 300	HD	4:3	10	30	200	900
480 X 360	HD	4:3	10	30	200	1000
640 X 480	FHD	4:3	10	30	250	1800
960 X 720	FHD	4:3	10	30	450	3000

表 5-37 不同场景下帧率和码率的推荐值

分辨率	推荐帧率	通信场景推荐码率	直播场景推荐码率
160 X 90	15	90	180
320 X 180	15	200	400
480 X 270	15	350	700
640 X 360	15	450	900
640 X 360	30	850	1700
800 X 450	15	700	1400
800 X 450	30	1050	2100
960 X 540	15	850	1700
960 X 540	30	1200	2400
1120 X 630	15	950	1900
1120 X 630	30	1400	2800
1280 X 720	15	1200	2400
1280 X 720	30	2000	4000
120 X 90	15	80	160

分辨率	推荐帧率	通信场景推荐码率	直播场景推荐码率
160 X 120	15	90	180
240 X 180	15	150	300
320 X 240	15	200	400
400 X 300	15	300	600
480 X 360	15	350	700
480 X 360	30	500	1000
640 X 480	15	600	1200
640 X 480	30	900	1800
960 X 720	15	1000	2000
960 X 720	30	1500	3000

HRTCLocalVideoStats

表 5-38 本地视频流信息详情

属性	类型	描述
width	int	视频宽
height	int	视频高
bitRate	int	视频码率
frameRate	int	视频帧率，单位：fps
packetLoss	int	视频丢包率
delay	int	时延，单位：ms
jitter	int	抖动
bytes	long	字节数
sendFrameRate	int	实际发送帧率，单位：fps

HRTCLocalAudioStats

表 5-39 本地音频流信息详情

属性	类型	描述
sampleRate	int	音频采样率
channels	int	音频频道数
sendVEL	int	发送语音电平
bitRate	int	音频码率
packetLoss	int	音频丢包率
delay	int	时延，单位：ms
jitter	int	抖动
bytes	long	字节数

HRTCRemoteVideoStats

表 5-40 远端视频流信息详情

属性	类型	描述
userId	String	远端用户userId
width	int	视频宽
height	int	视频高
bitRate	int	视频码率
frameRate	int	视频帧率，单位：fps
packetLoss	int	视频丢包率
delay	int	时延，单位：ms
jitter	int	抖动
bytes	long	字节数
rendererOutputFrameRate	int	渲染帧率，单位：fps
totalFrozenTime	int	远端用户在加入房间后到离开房间前，发生视频卡顿的累计时长，单位：ms
frozenRate	int	远端用户在加入房间后到离开房间前，发生视频卡顿的累计时长占视频总有效时长的百分比，单位：%

HRTCRemoteAudioStats

表 5-41 远端音频流信息详情

属性	类型	描述
userId	String	远端用户ID
sampleRate	int	音频采样率
channels	int	音频频道数
recvVEL	int	接收语音电平
bitRate	int	音频码率
packetLoss	int	音频丢包率
delay	int	时延，单位：ms
jitter	int	抖动
bytes	long	字节数
totalFrozenTime	int	远端用户在加入房间后到离开房间前，发生音频卡顿的累计时长，单位：ms
frozenRate	int	远端用户在加入房间后到离开房间前，发生音频卡顿的累计时长占音频总有效时长的百分比，单位：%

HRTCConnectInfo

表 5-42 跨房信息

属性	类型	描述
roomId	String	跨房房间号
role	HRTCRoleType	跨房时角色，具体请参见HRTCRoleType。

HRTCVideoFrame

表 5-43 视频帧

属性	类型	描述
format	HRTCVideoFrameFormat	支持的视频格式
videoData	byte[]	视频数据

属性	类型	描述
width	int	宽度，图像宽度，作为输入时，范围为[90,1920]，必须是4的整数倍
height	int	高度，图像高度，作为输入时，范围为[90,1200]，必须是2的整数倍

HRTCAudioFrame

表 5-44 音频帧

属性	类型	描述
frameType	HRTCAudioFrameType	音频格式
sampleRate	int	音频采样率
samplesPerSec	int	每秒采样数
bytesPerSample	int	每个采样点占用字节数
channels	int	声道数
data	byte[]	音频数据
buffer	ByteBuffer	音频数据

HRTCFrameBuffer

表 5-45 媒体帧数据

属性	类型	描述
mediaType	HRTCMediaType	媒体帧类型
buffer	ByteBuffer	媒体数据

HRTCLogLevel

表 5-46 日志级别

枚举值	描述
HRTC_LOG_LEVEL_ERROR	错误级别日志

枚举值	描述
HRTC_LOG_LEVEL_WARNING	警告级别日志
HRTC_LOG_LEVEL_INFO	信息级别日志
HRTC_LOG_LEVEL_DEBUG	调试级别日志

HRTCMediaType

表 5-47 媒体类型

枚举值	描述
HRTC_MEDIA_TYPE_AUDIO	音频，暂不支持
HRTC_MEDIA_TYPE_VIDEO	音频+视频

HRTCSpeakerModel

表 5-48 声音播放模式

枚举值	描述
HRTC_SPEAKER_MODE_EARPIECE	听筒模式
HRTC_SPEAKER_MODE_SPEAKER	外放模式

HRTCAudioRoute

表 5-49 音频路由

枚举值	描述
HRTC_AUDIO_ROUTE_SPEAKER	外放模式
HRTC_AUDIO_ROUTE_BLUETOOTH	蓝牙模式
HRTC_AUDIO_ROUTE_RECEIVER	听筒模式

枚举值	描述
HRTC_AUDIO_ROUTE_HEADSET	耳机模式

HRTCStreamType

表 5-50 流类型

枚举值	描述
HRTC_STREAM_TYPE_LD	流畅
HRTC_STREAM_TYPE_SD	标清
HRTC_STREAM_TYPE_HD	高清
HRTC_STREAM_TYPE_FHD	全高清

HRTCRoleType

表 5-51 用户角色

枚举值	描述
HRTC_ROLE_TYPE_JOINER	双向流角色，例如主播加入
HRTC_ROLE_TYPE_PLAYER	接收流角色，例如观众

HRTCVideoDisplayMode

表 5-52 图像填充模式

枚举值	描述
HRTC_VIDEO_DISPLAY_MODE_FIT	（不拉伸）黑边模式，通过扩边的方式保持宽高比。
HRTC_VIDEO_DISPLAY_MODE_HIDDEN	（不拉伸）裁剪模式，通过裁剪的方式保持宽高比。
HRTC_VIDEO_DISPLAY_MODE_FILL	视频尺寸进行缩放和拉伸以充满显示视窗。

HRTCConnStateTypes

表 5-53 网络连接状态

枚举值	描述
HRTC_CONN_DISCONNECTED	连接断开
HRTC_CONN_CONNECTING	建立网络连接中
HRTC_CONN_CONNECTED	网络连接成功
HRTC_CONN_RECONNETING	重新建立网络连接中
HRTC_CONN_FAILED	网络连接失败
HRTC_CONN_LOST	网络连接异常
HRTC_CONN_INTERRUPTED	网络连接中断

HRTCConnChangeReason

表 5-54 网络连接状态变化原因

枚举值	描述
HRTC_CONN_CHANGED_CONNECTING	正在连接
HRTC_CONN_CHANGED_JOIN_SUCCESS	加入房间成功
HRTC_CONN_CHANGED_RECONNECTING	重连中
HRTC_CONN_CHANGED_RECONNECT_SUCCESS	重连成功
HRTC_CONN_CHANGED_JOIN_FAILED	加入房间失败
HRTC_CONN_CHANGED_RECONNCET_FAILED	重连失败
HRTC_CONN_CHANGED_INTERRUPTED	连接中断
HRTC_CONN_CHANGED_KEEP_ALIVE_TIMEOUT	心跳超时

枚举值	描述
HRTC_CONN_CHANGED_LEAVE_ROOM	主动离开房间
HRTC_CONN_CHANGED_JOIN_ROOM_SERVER_ERROR	服务器异常
HRTC_CONN_CHANGED_SFU_BREAKDOWN	sfu服务故障
HRTC_CONN_CHANGED_JOIN_ROOM_AUTH_FAILED	鉴权失败，appId或者签名错误
HRTC_CONN_CHANGED_JOIN_ROOM_AUTH_RETRY	鉴权失败，重试
HRTC_CONN_CHANGED_JOIN_ROOM_AUTH_CLOCK_SYNC	鉴权时间戳校验失败
HRTC_CONN_CHANGED_JOIN_ROOM_URL_NOT_RIGHT	URL错误 400
HRTC_CONN_CHANGED_JOIN_ROOM_SERVICE_UNREACHABLE	服务不可达503
HRTC_CONN_CHANGED_INTERNAL_ERROR	内部错误
HRTC_CONN_CHANGED_KICKED_OFF	被踢出房间
HRTC_CONN_CHANGED_SIGNATURE_EXPIRED	签名过期
HRTC_CONN_REASON_USER_REMOVED	用户移除
HRTC_CONN_REASON_ROOM_DISMISSED	房间解散
HRTC_CONN_CHANGED_REGION_NOT_COVERED	区域未覆盖，所在区域不能提供SparkRTC服务。
HRTC_CONN_CHANGED_LOST	连接异常

HRTCRotationType

表 5-55 旋转类型

枚举值	描述
HRTC_ROTATION_TYPE_0	不旋转
HRTC_ROTATION_TYPE_90	顺时针旋转90度
HRTC_ROTATION_TYPE_180	顺时针旋转180度
HRTC_ROTATION_TYPE_270	顺时针旋转270度

HRTCVideoFrameFormat

表 5-56 视频帧格式

枚举值	描述
HRTC_VIDEO_FRAME_FORMAT_YUV420P	YUV420格式
HRTC_VIDEO_IMAGE_FORMAT_RGBA	RGBA格式
HRTC_VIDEO_IMAGE_FORMAT_2D	texture2d格式

HRTCAudioFrameType

表 5-57 音频帧格式

枚举值	描述
HRTC_AUDIO_FRAME_TYPE_PCM16	PCM 16位

HRTCVideoImageBufferType

表 5-58 视频帧缓冲区类型

枚举值	描述
HRTC_VIDEO_IMAGE_BUFFER_BYTE_ARRAY	Array类型，对应**HRTCVideoFrameFormat**中的YUV、RGBA格式

HRTCLeaveReason

表 5-59 离开房间原因

枚举值	描述
HRTC_LEAVE_REASON_USER_LEAVE_ROOM	用户主动离开
HRTC_LEAVE_REASON_SERVER_ERROR	服务器异常
HRTC_LEAVE_REASON_BREAKDOWN	sfu服务故障
HRTC_LEAVE_REASON_SERVICE_UNREACHABLE	服务不可达
HRTC_LEAVE_REASON_INTERNAL_ERROR	内部错误
HRTC_LEAVE_REASON_KICKED_OFF	被踢
HRTC_LEAVE_REASON_SIGNATURE_EXPIRED	签名过期
HRTC_LEAVE_REASON_RECONNECT_FAILED	重连超时

HRTCVideoMirrorType

表 5-60 视频镜像类型

枚举值	描述
HRTC_VIDEO_MIRROR_TYPE_AUTO	SDK决定镜像方式：前置摄像头镜像，后置摄像头不镜像
HRTC_VIDEO_MIRROR_TYPE_ENABLE	前置摄像头和后置摄像头都镜像
HRTC_VIDEO_MIRROR_TYPE_DISABLE	前置摄像头和后置摄像头都不镜像

HRTCRemoteAudioStreamStateReason

表 5-61 远端音频状态变化原因

枚举值	描述
HRTC_REMOTE_AUDIO_REASON_REMOTE_OFFLINE	远端用户离线
HRTC_REMOTE_AUDIO_REASON_REMOTE_MUTED	远端用户停止音频流发送
HRTC_REMOTE_AUDIO_REASON_REMOTE_UNMUTED	远端用户开启音频流发送
HRTC_REMOTE_AUDIO_REASON_REMOTE_FIRST_DECODED	远端用户音频解码第一帧

HRTCRemoteAudioStreamState

表 5-62 远端音频状态

枚举值	描述
HRTC_REMOTE_VIDEO_STATE_STOPPED	远端音频流关闭发送
HRTC_REMOTE_AUDIO_STATE_STARTING	远端音频流开启发送
HRTC_REMOTE_AUDIO_STATE_FIRST_DECODED	远端音频流解码第一帧

HRTCRemoteVideoStreamState

表 5-63 远端视频状态

枚举值	描述
HRTC_REMOTE_VIDEO_STATE_STOPPED	远端视频流关闭发送
HRTC_REMOTE_VIDEO_STATE_STARTING	远端视频流开启发送

HRTCRemoteVideoStreamStateReason

表 5-64 远端视频状态变化原因

枚举值	描述
HRTC_REMOTE_VIDEO_REASON_REMOTE_OFFLINE	远端用户离线
HRTC_REMOTE_VIDEO_REASON_REMOTE_MUTED	远端用户停止视频流发送
HRTC_REMOTE_VIDEO_REASON_REMOTE_UNMUTED	远端用户开启视频流发送
HRTC_REMOTE_VIDEO_REASON_LOCAL_MUTED	本端已取消选看远端视频流
HRTC_REMOTE_VIDEO_REASON_LOCAL_UNMUTED	本端已开启选看远端视频流

HRTCVolumeInfo

表 5-65 发言人音量

属性	类型	描述
roomId	String	房间ID
userId	String	用户ID
volume	int	音量

HRTCNetworkTestConfig

表 5-66 网络探测参数

属性	类型	描述
userId	String	必选，用户ID
roomId	String	房间ID，必选，建议值：userId+随机数拼接
signature	String	必选，签名鉴权
ctime	long	必选，时间戳
enableUplinkTest	boolean	必选，是否开启上行流探测

属性	类型	描述
enableDownlinkTest	boolean	必选，是否开启上行流探测
expectedUplinkBitrate	int	必选，用户期望的最高发送码率，单位为bps，范围为0以及 [100000, 5000000]，设为0表示由SDK指定最高码率
expectedDownlinkBitrate	int	必选，用户期望的最高接收码率，单位为bps，范围为0以及 [100000, 5000000]，设为0表示由SDK指定最高码率

HRTCNetworkTestResult

表 5-67 网络探测结果数据

属性	类型	描述
testState	**HRTCNetworkTestState**	测试结果
uplinkResult	**HRTCNetworkTestResultParam**	上行探测结果
downlinkResult	**HRTCNetworkTestResultParam**	下行探测结果

HRTCNetworkQualityLevel

网络质量级别的数值越大，信号越好。

表 5-68 网络质量级别

枚举值	描述
HRTC_NETWORK_QUALITY_UNKNOWN	网络质量未知
HRTC_NETWORK_QUALITY_EXCELLENT	网络质量非常好
HRTC_NETWORK_QUALITY_GOOD	网络质量好
HRTC_NETWORK_QUALITY_POOR	网络质量一般

枚举值	描述
HRTC_NETWORK_QUALITY_BAD	网络质量差
HRTC_NETWORK_QUALITY_VBAD	网络质量非常差

HRTCNetworkTestState

表 5-69 网络探测结果状态

枚举值	描述
HRTC_NETWORK_TEST_OK	探测成功
HRTC_NETWORK_TEST_FAIL	探测失败

HRTCNetworkTestResultParam

表 5-70 网络探测结果参数

属性	类型	描述
bitRate	int	探测的带宽
packetLoss	int	探测的丢包
delay	int	探测的时延
jitter	int	探测的抖动

HRTCMediaDirection

表 5-71 媒体方向指示

枚举值	描述
HRTC_MEDIA_LOCAL	本地媒体流
HRTC_MEDIA_REMOTE	远端媒体流

HRTCVideoStreamType

表 5-72 视频流类型

枚举值	描述
HRTC_VIDEO_STREAM_TYPE_BIG	视频大流
HRTC_VIDEO_STREAM_TYPE_SMALL	视频小流

HRTCVideoEncodeResolutionMode

表 5-73 视频编码分辨率比例模式

属性	描述
HRTC_VIDEO_ENCODE_RESOLUTION_MODE_NONE	不固定比例
HRTC_VIDEO_ENCODE_RESOLUTION_MODE_CONST_RATIO	固定比例

HRTCStreamPacketInfo

表 5-74 收包流的信息

属性	类型	描述
userId	String	用户ID
recvPacketCount	long	收包数，订阅的用户视频流收包数一直累加，重新订阅后数量清零
isAux	boolean	是否辅流

HRTCQualityInfo

表 5-75 网络质量信息

属性	类型	描述
userId	String	用户ID
width	int	宽度
height	int	高度

属性	类型	描述
level	HRTCNetworkQualityLevel	网络质量级别
mediaType	HRTCMediaType	媒体流类型

HRTCLocalAudioStreamStateReason

表 5-76 本地音频状态变化原因

枚举值	描述
HRTC_LOCAL_AUDIO_REASON_ERROR_OK	本地音频流状态正常
HRTC_LOCAL_AUDIO_REASON_ERROR_FAILURE	本地音频流出错原因不明确
HRTC_LOCAL_AUDIO_REASON_ERROR_RECORD_FAILURE	本地音频流录制失败，建议您检查录制设备是否正常工作
HRTC_LOCAL_AUDIO_REASON_ERROR_STOP_FAILURE	关闭采集失败
HRTC_LOCAL_AUDIO_REASON_ERROR_ACCESS_DENIED	音频设备无法访问，可能是设备隐私权限设置问题
HRTC_LOCAL_AUDIO_REASON_ERROR_ON_EXCLUSIVE_MODE	音频设备处于独占模式，且被其他应用独占，可以通知用户取消独占模式
HRTC_LOCAL_AUDIO_REASON_ERROR_ENDPOINT_CREATE_FAILED	音频设备终端创建失败，音频设备被拔出，或者已重新配置，禁用，删除了音频硬件或关联的硬件资源不可用。使用其他音频设备，重启或者更新驱动(仅适用于windows)
HRTC_LOCAL_AUDIO_REASON_ERROR_MMSYSERR_INVALPARAM	音频设备API非法参数，目前已知是杀毒软件导致(仅适用于windows)
HRTC_LOCAL_AUDIO_REASON_ERROR_MMSYSERR_NODRIVER	音频设备API返回无驱动，需要用户升级驱动(仅适用于windows)
HRTC_LOCAL_AUDIO_REASON_ERROR_AUDIO_SERVER_NOT_RUNNING	用户windows audio服务未启动，或者启动失败(仅适用于windows)
HRTC_LOCAL_AUDIO_REASON_ERROR__NO_DEVICE	没有设备(仅适用于windows)

枚举值	描述
HRTC_LOCAL_AUDIO_REASON_ERROR__RESTART_FAILED	扬声器播放无数据，重启失败

HRTCLocalAudioStreamState

表 5-77 本地音频状态

枚举值	描述
HRTC_LOCAL_AUDIO_STATE_STOPPED	本地音频流默认初始状态
HRTC_LOCAL_AUDIO_STATE_RECORDING	本地音频流录制设备启动成功
HRTC_LOCAL_AUDIO_STATE_FAILED	本地音频流启动失败

HRTCLocalVideoStreamState

表 5-78 本地视频状态

枚举值	描述
HRTC_LOCAL_VIDEO_STATE_STOPPED	本地视频流默认初始状态
HRTC_LOCAL_VIDEO_STATE_CAPTURING	本地视频流采集设备启动成功
HRTC_LOCAL_VIDEO_STATE_FAILED	本地视频流启动失败

HRTCLocalVideoStreamStateReason

表 5-79 本地视频状态变化原因

枚举值	描述
HRTC_LOCAL_VIDEO_REASON_ERROR_OK	本地视频流状态正常
HRTC_LOCAL_VIDEO_REASON_ERROR_FAILURE	本地视频流出错原因不明确

枚举值	描述
HRTC_LOCAL_VIDEO_REASON_ERROR_CAPTURE_FAILURE	本地视频流录制失败，建议您检查录制设备是否正常工作
HRTC_LOCAL_VIDEO_REASON_ERROR_STOP_FAILURE	关闭采集失败
HRTC_LOCAL_VIDEO_REASON_ERROR_CAPTURE_DEVICE_NO_PERMISSION	没有摄像头权限
HRTC_LOCAL_VIDEO_REASON_ERROR_CAPTURE_DEVICE_BUSY	摄像头设备已占用
HRTC_LOCAL_VIDEO_REASON_ERROR_CAPTURE_APP_IN_BACKGROUND	应用处于后台
HRTC_LOCAL_VIDEO_REASON_ERROR_CAPTURE_OPEN_CAMERA_FAILED	打开摄像头设备失败
HRTC_LOCAL_VIDEO_REASON_ERROR_CAPTURE_MULTIPLE_FOREGROUND_APP	应用窗口处于侧拉、分屏、画中画模式 (仅适用于iOS)
HRTC_LOCAL_VIDEO_REASON_ERROR_CAPTURE_DEVICE_DISCONNECTED	本地视频采集设备未连接 (仅适用windows和macOS)

HRTCRemoteAudioMode

表 5-80 视频编码分辨率比例模式

属性	描述
RTC_REMOTE_AUDIO_SUBSCRIBED	订阅模式，接收所有已订阅远端声音
RTC_REMOTE_AUDIO_TOP_THREE	TopN模式，接收远端声音最大三路音频
HRTC_REMOTE_AUDIO_P2P	P2P模式
HRTC_REMOTE_AUDIO_RTSA_CMD	RTSA-CMD模式

HRTCMediaConnStateTypes

表 5-81 媒体连接状态类型

属性	描述
HRTC_MEDIA_CONN_CONNECTED	与媒体服务器连接成功
HRTC_MEDIA_CONN_FAILED	与媒体服务器建链失败

HRTCMediaConnChangeReason

表 5-82 媒体连接状态变化原因

属性	描述
HRTC_MEDIA_CONN_CHANGED_CONNECTED	连接成功
HRTC_MEDIA_CONN_CHANGED_NAT_FAILED	与媒体服务器NAT未打通

HRTCImageBufferFormat

表 5-83 视频帧图片格式

属性	类型	描述
format	**HRTCVideoFrameFormat**	视频帧图片存储格式
bufferType	**HRTCVideoImageBufferType**	视频帧缓冲区存储类型

HRTCVideoAuxiliaryEncParam

表 5-84 辅流编码参数

属性	类型	描述
width	int	宽度
height	int	高度
frameRate	int	帧率
bitrate	int	码率

HRTCEngineConfig

表 5-85 引擎创建相关参数

属性	类型	描述
context	Context	上下文
appId	String	应用ID，只有App ID相同的应用程序才能进入同一个房间进行互动。appId获取方法请参见**应用管理**
countryCode	String	国家码，具体请参见**国家码对照表**
logEnable	boolean	日志是否输出到文件，true为输出到文件，false为不输出到文件
logLevel	HRTCLogLevel	日志等级，取值为HRTC_LOG_LEVEL_ERROR、HRTC_LOG_LEVEL_WARNING、HRTC_LOG_LEVEL_INFO或HRTC_LOG_LEVEL_DEBUG，默认为HRTC_LOG_LEVEL_DEBUG
logPath	String	日志路径，需调用方保证路径合法可用，rtc仅做基础校验
logSize	int	日志大小，单位为字节，目前固定10*1024

HRTCAreaCode

表 5-86 访问区域

枚举值	描述
HRTC_AREA_CODE_GLOB	全球（默认）
HRTC_AREA_CODE_CN	中国
HRTC_AREA_CODE_NA	中北美
HRTC_AREA_CODE_SA	拉美
HRTC_AREA_CODE_EU	欧洲
HRTC_AREA_CODE_SEA	东南亚
HRTC_AREA_CODE_AF	非洲
HRTC_AREA_CODE_AS	亚洲

HRTCOnStats

表 5-87 引擎创建相关参数

属性	类型	描述
cpuAppUsage	double	app的cpu利用率，单位(%)
cpuTotalUsage	double	cpu总利用率，单位(%)
memoryAppUsageInKbytes	int	app占用内存，单位KB
memoryAppUsageRatio	double	app占用的内存率，单位(%)
memoryTotalUsageRatio	double	总的内存利用率，单位(%)
gatewayRtt	int	到本地网关的延迟，单位ms
sendBytes	long	总的发送字节数，单位bytes
sendVideoBytes	long	视频的发送字节数，单位bytes
sendAudioBytes	long	音频的发送字节数，单位bytes
receiveBytes	long	总的接收字节数，单位bytes
receiveVideoBytes	long	视频的接收字节数，单位bytes
receiveAudioBytes	long	音频的接收字节数，单位bytes
sendBitRate	int	总的发送比特率，单位Kbps
sendVideoBitRate	int	视频的发送比特率，单位Kbps
sendAudioBitRate	int	音频的发送比特率，单位Kbps
receiveBitRate	int	总的接收比特率，单位Kbps
receiveVideoBitRate	int	视频的接收比特率，单位Kbps
receiveAudioBitRate	int	音频的接收比特率，单位Kbps
sendLossRate	int	发送丢包率，单位(%)
receiveLossRate	int	接收丢包率，单位(%)

属性	类型	描述
lastmileDelay	int	到服务器的延迟，单位ms

HRTCAudioFileState

表 5-88 音频文件播放状态

属性	描述
HRTC_AUDIO_FILE_OPEN_COMPLETED	成功打开音频文件
HRTC_AUDIO_FILE_OPENING	正在打开音频文件
HRTC_AUDIO_FILE_IDLE	音频文件播放就绪
HRTC_AUDIO_FILE_PLAYING	音频文件播放中
HRTC_AUDIO_FILE_PLAY_COMPLETED	音频文件播放完成
HRTC_AUDIO_FILE_PAUSED	音频文件暂停播放
HRTC_AUDIO_FILE_STOPPED	音频文件停止播放
HRTC_AUDIO_FILE_FAILED	音频文件播放失败
HRTC_AUDIO_FILE_POSITION_UPDATE	音频文件播放进度更新
HRTC_AUDIO_FILE_STATE_UNKNOWN	音频文件播放状态未知

HRTCAudioFileReason

表 5-89 音频播放状态变化原因

枚举值	描述
HRTC_AUDIO_FILE_REASON_NONE	没有错误
HRTC_AUDIO_FILE_REASON_URL_NOT_FOUND	未找到URL
HRTC_AUDIO_FILE_REASON_CODEC_NOT_SUPPORTED	解码器不支持该编码
HRTC_AUDIO_FILE_REASON_INVALID_ARGUMENTS	非法参数
HRTC_AUDIO_FILE_REASON_SRC_BUFFER_UNDERFLOW	播放缓冲区数据不足
HRTC_AUDIO_FILE_REASON_INTERNAL	内部错误

枚举值	描述
HRTC_AUDIO_FILE_REASON_INVALID_STATE	播放器内部状态错误
HRTC_AUDIO_FILE_REASON_NO_RESOURCE	没有该资源
HRTC_AUDIO_FILE_REASON_OBJ_NOT_INITIALIZED	对象未初始化
HRTC_AUDIO_FILE_REASON_INVALID_CONNECTION_STATE	播放器与服务器连接无效
HRTC_AUDIO_FILE_REASON_UNKNOWN_STREAM_TYPE	未知的媒体流类型
HRTC_AUDIO_FILE_REASON_VIDEO_RENDER_FAILED	渲染失败
HRTC_AUDIO_FILE_REASON_INVALID_MEDIA_SOURCE	无效的媒体资源
HRTC_AUDIO_FILE_REASON_UNKNOWN	状态未知

HRTCMediaOptions

表 5-90 媒体选参

属性	类型	描述
autoSubscribeAudio	bool	自动订阅远端音频
autoSubscribeVideo	bool	自动订阅远端视频

HRTCEngineContext

表 5-91 引擎初始化参数

属性	类型	描述
engineConfig	HRTCEngineConfig	引擎配置项，具体请参见 **HRTCEngineConfig**
logConfig	HRTCLogConfig	日志配置项，具体请参见**HRTCLogConfig**
eventHandler	IHRTCEngineEventHandler	事件回调，具体请参见 **IHRTCEngineEventHandler**

HRTCLogConfig

表 5-92 日志参数

属性	类型	描述
level	HRTCLogLevel	日志级别，具体请参见**HRTCLogLevel**
path	const char *	日志路径
logSize	int	日志大小
enable	bool	日志开关

HRTCSfuType

表 5-93 Sfu 类型

枚举值	描述
HRTC_SFU_TYPE_PUBLIC_NETWORK	公网sfu资源

HRTCDeviceState

表 5-94 系统音视频设备状态

枚举值	描述
HRTC_DEVICE_STATE_ACTIVE	激活状态，设备可用
HRTC_DEVICE_STATE_DISABLED	设备禁用
HRTC_DEVICE_STATE_UNPLUGGED	设备拔出

HRTCUrlStatusList

表 5-95 HRTCRtmpUrlInfo

属性	描述
char url[1025]	url字符串
int status	状态码
int errCode	错误码

HRTCVideoRemoteView

表 5-96 远端流视图

属性	描述
SurfaceView view	窗口句柄
HRTCStreamType streamType	流模式（LD/SD/HD/FHD/THD）
String userId	用户ID
boolean disableAdjustRes	是否自适应
HRTCStreamType minResolution	自适应场景下，建议的最低选择档位

HRTCNetWorkBandwidth

表 5-97 HRTCNetWorkBandwidth

属性	描述
int maxBandwidth	网络带宽上限，有效范围为[3072,51200]，即3M~50M。

HRTCRemoteMicState

表 5-98 远端麦克风设备状态

枚举值	描述
HRTC_REMOTE_MIC_STATE_UNMUTE	麦克风设备状态正常
HRTC_REMOTE_MIC_STATE_MUTE	麦克风设备状态静音

HRTCMultiRoomMediaRelayConfiguration

表 5-99 HRTCMultiRoomMediaRelayConfiguration

属性	类型	描述
srcRoomMediaInfo	**HRTCSrcMultiRoomMediaInfo**	源房间的鉴权信息

属性	类型	描述
destRoomMediaInfo	HRTCDstMultiRoomMediaInfo	目的跨房的房间信息以及鉴权信息

HRTCSrcMultiRoomMediaInfo

表 5-100 HRTCSrcMultiRoomMediaInfo

属性	类型	描述
authorization	const char*	源房间的鉴权信息
userId	const char*	源房间的用户名(必须为0)
roomId	const char*	源房间的房间号
ctime	long long	鉴权时间信息

HRTCDstMultiRoomMediaInfo

表 5-101 HRTCDstMultiRoomMediaInfo

属性	类型	描述
authorization	const char*	目标跨房的鉴权信息
userId	const char*	目标跨房的虚拟用户名
roomId	const char*	目标跨房房间号
userRole	HRTCRoleType	跨房角色
ctime	long long	鉴权时间信息

HRTCMultiRoomMediaRelayState

表 5-102 HRTCMultiRoomMediaRelayState

属性	描述
HRTC_MULTI_ROOM_MEDIA_RELAY_STATE_IDLE	就绪状态
HRTC_MULTI_ROOM_MEDIA_RELAY_STATE_CONNECTING	正在连接

属性	描述
HRTC_MULTI_ROOM_MEDIA_RELAY_STATE_RUNNING	主播成功加入目标房间
HRTC_MULTI_ROOM_MEDIA_RELAY_STATE_FAILURE	发生异常

HRTCMultiRoomMediaRelayStateCode

表 5-103 HRTCMultiRoomMediaRelayStateCode

属性	描述
HRTC_MULTI_ROOM_MEDIA_RELAY_OK	正常状态
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_SERVER_NO_RESPONSE	服务端无响应
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_INTERNAL_ERROR	服务器内部出错
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_USER_OVER_LIMIT	用户跨房超出限制数
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_OVER_LIMIT	房间跨房用户超出限制数
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_REQ_EMPTY	跨房请求消息体为空
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_OPERATION_CONFLICT	跨房请求，加入和退出存在冲突
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_SRC_USERINFO_INVALID	跨房请求原用户信息无效
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_WITH_ORI	跨房房间与原用户房间相同
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_REPEAT	跨房请求房间重复
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_USER_EXISTED	跨房用户已存在
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_INVALID_REQUEST	无效请求
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_IS_NOT_EXIST	房间不存在
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_FRAME_TYPE_NOT_EQUAL	跨房源房间和目的房间加密模式不一致

属性	描述
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_AUTHENTICATION_FAILURE	鉴权失败
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_REMOVE_INFO_NOT_EXIST	退出跨房信息不存在
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_EXCEPTION_STOP	异常退出跨房

5.4.8 媒体原始数据管理

5.4.8.1 注册回调（IHRTCMediaEngine）

表 5-104 IHRTCMediaEngine

接口	描述
setVideoFrameObserver	注册原始视频媒体数据监听回调
setAudioFrameObserver	注册原始音频媒体数据监听回调
setEncDecryptFrameObserver	注册端到端加密App加密模式下的原始媒体数据回调

setVideoFrameObserver

```
public int setVideoFrameObserver(HRTCConnection conn, IHTCVideoFrameObserver observer);
```

【功能说明】

注册原始视频媒体数据监听回调。

【请求参数】

- conn：HRTCConnection引擎。
- observer：原始视频数据处理接口，具体请参见IHRTCVideoFrameObserver。

【返回参数】

- 0：表示调用命令成功。
- > 0：表示调用命令失败。

setAudioFrameObserver

```
public int setAudioFrameObserver(IHTCAudioFrameObserver observer);
```

【功能说明】

注册原始音频媒体数据监听回调。

【请求参数】

observer：原始音频数据处理接口，具体见**IHRTCAudioFrameObserver**。

【返回参数】

- 0：表示调用命令成功。
- > 0：表示调用命令失败。

setEncDecryptFrameObserver

```
public int setEncDecryptFrameObserver(IHRTCEncDecryptFrameObserver observer);
```

【功能说明】

注册端到端加密App加密模式下的原始媒体数据回调。

【请求参数】

observer：原始媒体数据回调，具体见**IHRTCEncDecryptFrameObserver**。

【返回参数】

- 0：表示调用命令成功。
- > 0：表示调用命令失败。

5.4.8.2 事件回调（IHRTCVideoFrameObserver）

表 5-105 视频事件回调说明

接口	描述
onVideoFrameCapture	原始视频回调（前处理)
onVideoFrameRender	渲染后视频回调（后处理）
requireCaptureVideoFrame	是否开启前处理
requireRenderVideoFrame	是否开启后处理

onVideoFrameCapture

```
void onVideoFrameCapture(HRTCVideoFrame videoFrame)
```

【功能说明】

原始视频回调，从接口回调中取到原始视频数据以作前处理。

【回调参数】

videoFrame：视频数据格式 ，具体请参见**HRTCVideoFrame**。

onVideoFrameRender

```
void onVideoFrameRender(String userId, HRTCVideoFrame videoFrame)
```

【功能说明】

原始视频数据处理后回调。

【回调参数】

- userid：用户ID。
- videoFrame：视频数据格式 ，具体请参见**HRTCVideoFrame**。

requireCaptureVideoFrame

```
boolean requireCaptureVideoFrame()
```

【功能说明】

是否需要开启前处理。

【返回参数】

- true：开启。
- false：不开启。

requireRenderVideoFrame

```
boolean requireRenderVideoFrame()
```

【功能说明】

是否需要开启后处理。

【返回参数】

- true：开启。
- false：不开启。

5.4.8.3 事件回调（IHRTCAudioFrameObserver）

表 5-106 音频事件回调说明

接口	描述
onAudioFramePlayback	音频播放回调（后处理）
onAudioFrameMixed	音频混音处理回调
onAudioFrameRecord	音频采集回调（前处理）
requirePlaybackAudioFrame	是否开启音频后处理
requireRecordAudioFrame	是否开启音频前处理
requireMixedAudioFrame	是否开启音频混音回调

onAudioFramePlayback

```
void onAudioFramePlayback(HRTCAudioFrame audioFrame)
```

【功能说明】

需要播放的音频数据回调，从接口回调中取到音频数据以作后处理。

【回调参数】

audioFrame：音频数据格式 ，具体请参见**HRTCAudioFrame**。

onAudioFrameMixed

```
void onAudioFrameMixed(HRTCAudioFrame audioFrame)
```

【功能说明】

全部音频混音数据回调，包含上下行所有通道。

【回调参数】

audioFrame：音频数据格式 ，具体请参见**HRTCAudioFrame**。

onAudioFrameRecord

```
void onAudioFrameRecord(HRTCAudioFrame audioFrame)
```

【功能说明】

音频采集原始数据回调，对音频数据的修改会发送到远端。

【回调参数】

audioFrame：音频数据格式 ，具体请参见**HRTCAudioFrame**。

requireRecordAudioFrame

```
boolean requirePlaybackAudioFrame()
```

【功能说明】

是否开启音频前处理。

【返回参数】

- true：开启。
- false：不开启。

requirePlaybackAudioFrame

```
boolean requirePlaybackAudioFrame()
```

【功能说明】

是否需要开启音频后处理。

【返回参数】

- true：开启。
- false：不开启。

requireMixedAudioFrame

```
boolean requireMixedAudioFrame()
```

【功能说明】

是否需要开启全部音频混音数据回调。

【返回参数】

- true：开启。
- false：不开启。

5.4.8.4 事件回调（IHRTCEncDecryptFrameObserver）

表 5-107 媒体数据回调说明

接口	描述
onMediaFrameEncrypt	编码后的媒体数据（加密处理）
onMediaFrameDecrypt	解码前的媒体数据（解密处理）

onMediaFrameEncrypt

```
void onMediaFrameEncrypt(HRTCFrameBuffer frameBuffer)
```

【功能说明】

媒体数据回调到app层加密处理。

【回调参数】

frameBuffer：媒体数据格式，具体请参见HRTCFrameBuffer。

【返回参数】

无

onMediaFrameDecrypt

```
void onMediaFrameDecrypt(HRTCFrameBuffer frameBuffer)
```

【功能说明】

媒体数据回调到app层解密处理。

【回调参数】

frameBuffer：媒体数据格式，具体请参见HRTCFrameBuffer。

【返回参数】

无

5.5 常见问题

- **初始化引擎时，countryCode应该如何填写?**
  countryCode表示区域码，如果对应的是GLOBAL类型，则直接填空字符串，即""，如果是CN类型，则可以填"CN"。具体请参见国家码对照表。
- **初始化引擎时，domain应该如何填写?**
  该字段已废弃，不需要再传值。
- **加入房间时，username必须填吗?**
  必填。
- **应用关闭鉴权时，加入房间还需要填鉴权参数吗?**
  不需要填。

- **加入房间失败时，如何解决?**

  首先通过返回的**客户端错误码**进行分析，主要有以下几个原因：

  - 网络问题，您需要确认网络是否正常运行。
  - 鉴权问题，应用默认开启鉴权，您需要确保鉴权生成正确，参数合理传入，且保证没过期，需要注意ctime参数的设置，具体可参考**接入鉴权**。
  - 参数问题，确认关键参数是否填写，以及是否正确填写，比如username是必填的，countryCode无特殊需求，则填空字符串。countryCode填写请参见**国家码对照表**。

- **如何进行屏幕共享?**

  可以参考屏幕共享场景，按照正确时序使用API。

- **如何进行视频自渲染显示正常图像?**

  在开启自渲染后，通过**onRenderExternalVideoFrame**接口接收SDK回调的图像信息，需要自行解析其中的图像信息，包括尺寸、格式、数据。以YUV格式举例，SDK返回的YUV格式是YUV420，故Y、U、V三分量需要按照下面的方式获取：

  ```
  ByteBuffer buffer = videoFrame.getBuffer();
  byte[] data = new byte[buffer.limit];
  buffer.get(data);

  byte[] yData = new byte[width * height];
  System.arraycopy(data, 0, yData, 0, width * height);

  byte[] uData = new byte[width * height / 4];
  System.arraycopy(data, width * height, uData, 0, width * height / 4);

  byte[] vData = new byte[width * height / 4];
  System.arraycopy(data, width * height * 5 / 4, vData, 0, width * height / 4);
  ```

  得到YUV数据后，可以选择OpenGL或者其他能够播放YUV格式图像的方式进行渲染

5.6 修订记录

表 5-108 修订记录

修改时间	修改说明
2022-06-21	第十八次正式发布 本次变更如下： ● 新增以下接口： – addMultiRoomMediaRelay：添加单个跨房 – removeMultiRoomMediaRelay：删除单个跨房 – stopMultiRoomMediaRelay：停止所有跨房 ● 新增以下事件回调： – onRemoteMicrophoneStateChanged：远端麦克风设备状态变更通知 – onUserNetworkQualityNotify：用户级网络质量回调 ● 新增以下数据类型： – HWRtcRemoteMicState：麦克风设备状态 – HRTCMultiRoomMediaRelayConfiguration：跨房配置 – HRTCSrcMultiRoomMediaInfo：源房间信息 – HRTCDstMultiRoomMediaInfo：目标房间信息 – HRTCMultiRoomMediaRelayState：跨房状态 – HRTCMultiRoomMediaRelayStateCode：跨房状态码
2022-03-24	第十七次正式发布 本次变更如下： 修改appid获取方式的相关描述。
2022-03-18	第十六次正式发布 本次变更如下： 新增setNetworkBandwidth接口
2022-02-25	第十五次正式发布 本次变更如下： 修改接口：HRTCVideoDisplayMode去掉自适应HRTC_VIDEO_DISPLAY_MODE_ADAPT
2021-12-02	第十四次正式发布 本次变更涉及部分API的描述优化，具体请参见**接口参考**。

修改时间	修改说明
2021-11-22	第十三次正式发布 本次变更涉及部分API的逻辑优化与融合，如joinRoom、onConnectionChangedNotify、pushLocalVideo等，具体请参见**接口参考**。
2021-08-13	第十二次正式发布 本次发布版本为1.10.0版本，整合了之前若干版本和分支的一个全新版本。 本次变更如下： 新增**joinRoom**、**adjustPlaybackVolume**接口。
2021-06-10	第十一次正式发布 本次发布版本为1.8.0版本，整合了之前若干版本和分支的一个全新版本。 本次变更如下： 新增的接口及回调： • **changeUserName**、onRemoteUserNameChangedNotify、onUserNameChangedNotify：会议中修改用户昵称的接口及本地和远端会收到的回调。 • **createConnection**：加入多房间（跨房）前，与要跨入的房间先建立连接的接口，跨一个房间建立一个连接。 • **setDefaultSpeakerModel**：设置默认的声音播放模式。 • **onMediaConnectStateChangedNotify**：增加与媒体服务器连接状态变化的通知回调。 • **onFirstRemoteAuxiliaryStreamDecoded**：引擎收到第一帧远端辅流并解码成功的回调。 新增了接口对象和对应事件回调： • **HRTCConnection** • **事件回调(IHRTCConnectionEventHandler)** 废弃的接口及回调： connectOtherRoom、onConnectOtherRoom、disconnectOtherRoom、onDisconnectOtherRoom：上述的两个接口及其对应的回调在1.8.0版本中予以废弃。新的跨房通过创建**HRTCConnection**与对应房间先建立连接，然后通过调用该对象的**joinRoom**和**leaveRoom**接口实现跨房和退房功能，对应回调处理通过**事件回调(IHRTCConnectionEventHandler)**中的**onJoinRoomSuccess**、**onJoinRoomFailure**、**onLeaveRoom**实现。

修改时间	修改说明
2021-01-28	第十次正式发布 本次变更如下： • **HRtcEngine**新增接口：getVersion
2020-12-23	第九次正式发布 本次变更如下： • **HRtcEngine**新增接口：updateLocalRenderMode、enableSmallVideoStream、setPriorRemoteVideoStreamType、setRemoteVideoStreamType、setRemoteVideoAdjustResolution、setupRemoteView、pullRemoteVideo、pullAllRemoteVideo、renewAuthorization、enableLocalAudioStream、 • 新增**事件回调**：onFirstRemoteVideoDecoded、onAuthorizationExpired • 新增**数据类型**：HRTCVideoStreamType、HRTCVideoEncodeResolutionMode • HRTCErrorCode增加90000043~90000045错误码
2020-11-26	第八次正式发布 本次变更如下： • 新增**数据类型**：RTCMediaDirection • 新增SparkRTC**接入鉴权**方法说明
2020-11-18	第七次正式发布 本次变更如下： **HRtcEngine**新增接口：setAuxiliaryExternalVideoFrameOutputEnable
2020-10-21	第六次正式发布 本次变更如下： • **HRtcEngine**新增接口：startNetworkTest、stopNetworkTest、enableStats • 新增**事件回调**：onUserVolumeStatsNotify、onStartAudioFileNotify、onStopAudioFileNotify、onPauseAudioFileNotify、onResumeAudioFileNotify、onNetworkTestQuality、onNetworkTestResult • 新增**数据类型**：HRTCVolumeInfo、HRTCNetworkTestConfig、HRTCNetworkTestResult、HRTCNetworkQualityLevel、HRTCNetworkTestState、HRTCNetworkTestResultParam

修改时间	修改说明
2020-09-04	第五次正式发布 本次变更如下： ● 增加视频镜像接口：，setVideoEncoderMirror ● 增加音视频流采集发流控制接口及流状态变化回调：enableLocalVideo，onRemoteAudioStateChangedNotify，onRemoteVideoStateChangedNotify ● 增加重连成功回调接口onRejoinRoomSuccess ● HRtcErrorCode增加90000040~90000042错误码 ● 增加HRTCLeaveReason，HRTCVideoMirrorType，HRTCRemoteAudioStreamState，HRTCRemoteAudioStreamStateReason，HRTCRemoteVideoStreamState，HRTCRemoteVideoStreamStateReason枚举
2020-08-17	第四次正式发布 本次变更如下： ● 增加选看跨房功能接口及回调：connectOtherRoom，disconnectOtherRoom，onConnectOtherRoom，onDisconnectOtherRoom ● 增加音频流接收选择接口：muteRemoteAudio，muteAllRemoteAudio ● 增加音视频流统计信息上报：onVideoStatsNotify，onAudioStatsNotify，onAuxiliaryStreamStatsNotify、onConnectOtherRoom、onDisconnectOtherRoom ● HRtcErrorCode增加90000034~90000039错误码 ● 服务端错误码增加RTC.32000030~RTC.32000033错误码 ● 增加HRTCVideoFrameFormat枚举类
2020-07-03	第三次正式发布 本次变更如下： ● 选看远端接口修改为startRemoteStreamView，updateRemoteRenderMode，stopRemoteStreamView ● 增加辅流相关接口startRemoteAuxiliaryStreamView，stopRemoteAuxiliaryStreamView，setRemoteAuxiliaryStreamViewRotation，updateRemoteAuxiliaryStreamRenderMode ● 修改错误码等

修改时间	修改说明
2020-06-20	第二次正式发布 本次变更如下： ● SDk集成中，修改需要添加的库文件。 ● 修改接口参考中的类、方法等内容，具体如下： – HRtcEngine类增加adjustRecordingVolume、adjustPlaybackVolume、setVideoEncoder、changeUserRole等方法增加。 – 新增结构体LocalVideoStats、RemoteVideoStats、StatsInfo、VideoEncoder。 – IHRtcEngineEventHandler类增加onJoinRoomFailure、onUserRoleChangedNotify、onAudioRouteStateChangedNotify和onVideoStatsNotify方法。 – RtcErrorCode增加90000019~90000030的错误码。 – 修改并增加服务端错误码。 – 增加RoleType枚举类。 – ConnChangeReason枚举类增加枚举值成员。
2020-04-15	第一次正式发布

6 iOS/macOS SDK

6.1 开发前准备

6.1.1 iOS 开发前准备

前提条件

已**提交工单**获取SDK包。

环境要求

- 准备XCODE集成开发。
- 准备iOS 8.0及以上的iPhone真机。
- 支持的终端CPU架构：arm64，arm32。

SDK 集成

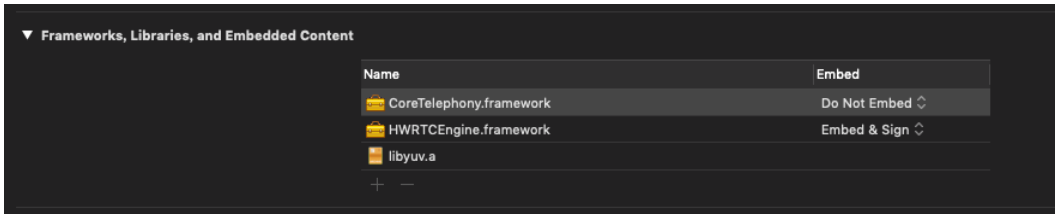
步骤1 解压iOS SDK包。

步骤2 将iOS SDK包中的HWRTCEngine动态库和RTC.bundle文件导入创建的XCODE工程中。

图 6-1 导入 HWRTCEngine 动态库和 RTC.bundle 文件

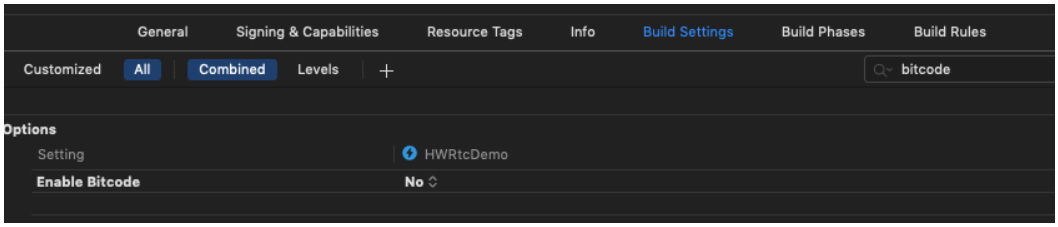
步骤3 在“General”页面将SDK中HWRTCEngine.framework文件加入到工程。如果需要使用混音功能，请添加hwffmpeg.framework库文件至HWRTCEngine.framework同级目录。

图 6-2 添加 HWRTCEngine.framework



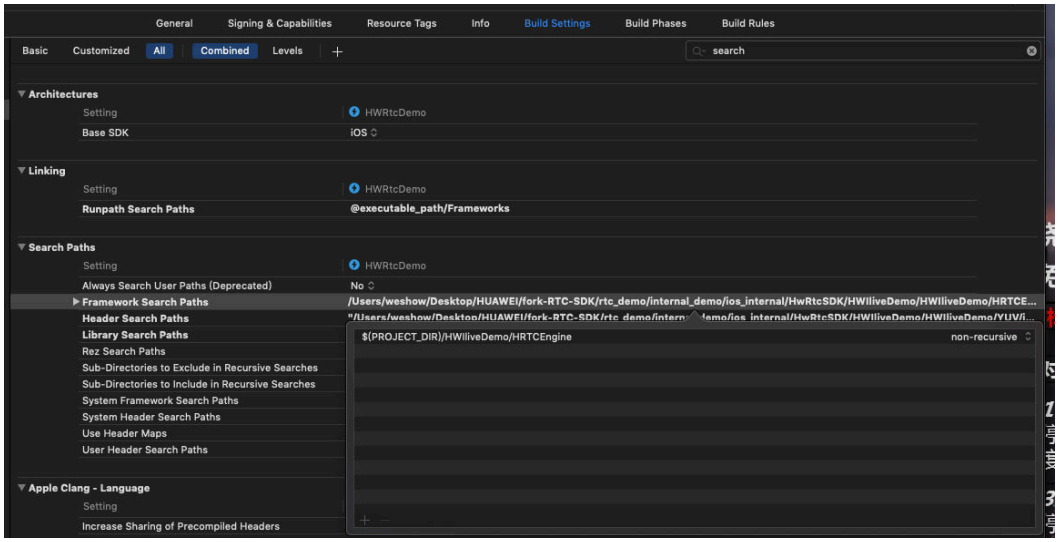
步骤4 在“Build Settings”页面关闭“Enable Bitcode”， 将其设置为“No”。

图 6-3 设置 Enable Bitcode



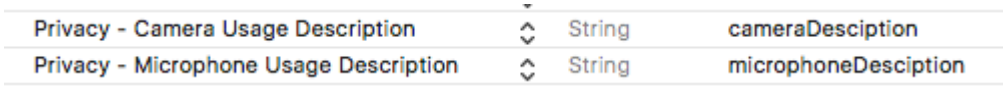
步骤5 在“Build Settings”页面的搜索框输入“search”，查看Framework search paths路径是否正确，确保文件加载成功。

图 6-4 检查文件是否加载成功



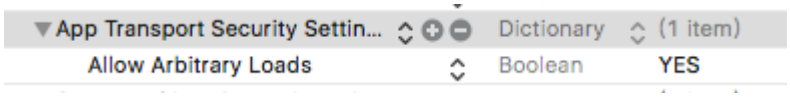
步骤6 在“info.plist”文件中增加摄像头和麦克风权限。

图 6-5 摄像头和麦克风权限



步骤7 在“info.plist”文件中添加ATS。

图 6-6 添加 ATS



步骤8 使用XCODE连接iPhone，编译工程，若界面提示“Build Success”，则完成SDK集成。

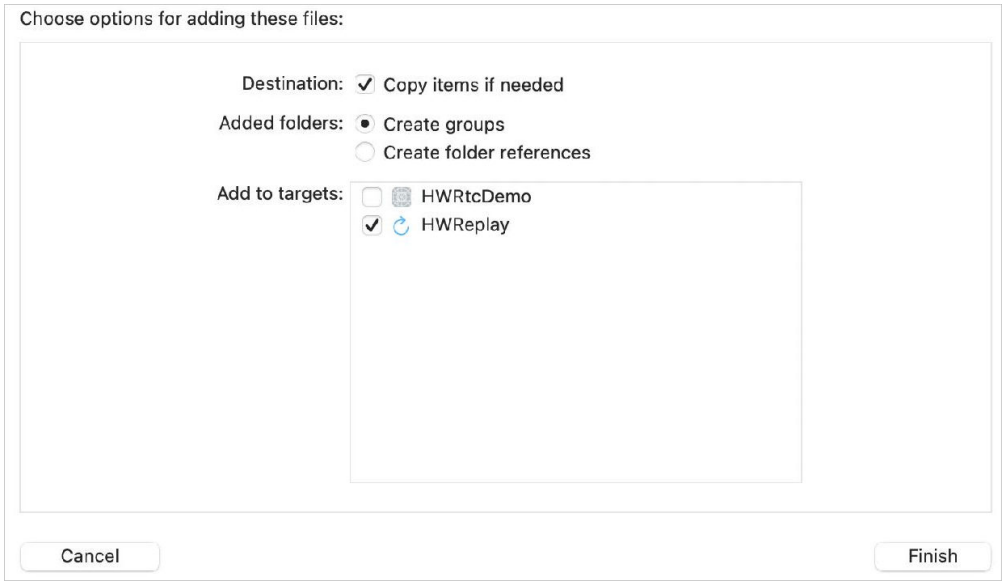
----结束

HWRtcEngineReplayKit 集成（可选）

若您需要使用**屏幕共享**功能，则需要导入**HWRtcEngineReplayKit.framework**。在导入HWRtcEngineReplayKit.framework前，您需要参考**屏幕共享**完成Broadcast Upload Extension的创建。

步骤1 在xcode项目中的Broadcast Upload Extension中导入HWRtcEngineReplayKit.framework。

图 6-7 导入文件

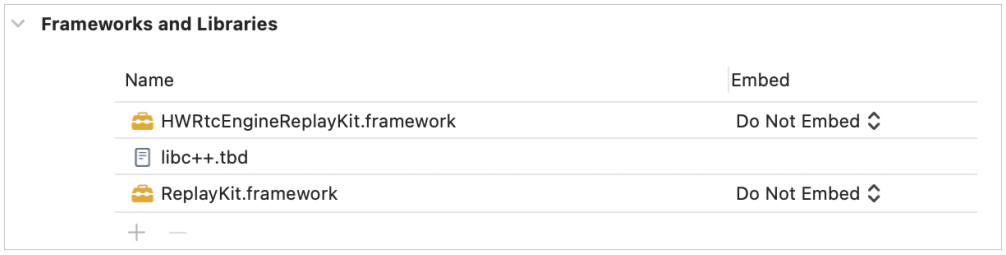


步骤2 在xcode项目中的Broadcast Upload Extension中导入libc++.tbd库。

图 6-8 导入库

导入成功后，会在Broadcast Upload Extension中的Framworks and libraries中查看到HWRtcEngineReplayKit.framework，如图6-9所示。

图 6-9 查看导入结果



----结束

6.1.2 macOS 开发前准备

前提条件

已**提交工单**获取SDK包。

环境要求

- 准备XCODE集成开发环境。
- 准备MAC设备，支持macOS 10.11以上的设备。
- 支持的终端CPU架构：x86_64。

SDK 集成

支持dylib和framework两种包集成。

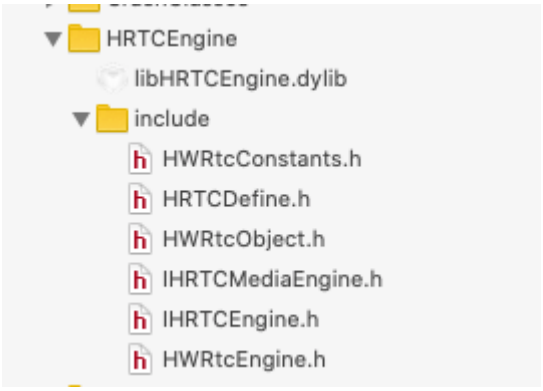
步骤1 解压Mac SDK包。

步骤2 将MAC SDK包中的lib动态库libHRTCEngine.dylib或者libHRTCEngine.framework和RTC.bundle文件导入创建的XCODE工程中，如果用到混音功能，需要将hwffmpeg.framework至libHWRTCEngine.framework同级目录。

图 6-10 lib 库

步骤3 将MAC SDK包中的头文件“HWRtcObject.h”、“HWRtcConstants.h”、“HRTCDefine.h”“IHRTCMediaEngine.h”、“IHRTCEngine.h”和“HWRtcEngine.h”导入工程中，集成framework在Headers下面自带头文件，无需再导入。

图 6-11 导入头文件



步骤4 在“Build Settings”页面的搜索框输入“search”，确保头文件和库文件的位置都已经在XCODE设置成功。

若“Header Search Paths ”和“Library Search Paths”中文件位置不对，可以将XCODE对应文件夹直接拖过来即可。

图 6-12 检查文件是否加载成功

步骤5 编译工程，若界面提示“Build Success”，则完成SDK集成。

----结束

6.2 SDK 使用

步骤1 创建引擎并初始化。

域名不需要设置，由SDK自动获取。**appId**获取方法请参见**创建应用**。

```
_rtcEngine = [HWRtcEngine sharedEngine];
HWRtcEngineConfig * cfg = [[HWRtcEngineConfig alloc]init];
cfg.appId = appid;// appId需在控制台中创建应用后获取
cfg.domain = domain;// 该字段已废弃，不需要再传值
cfg.countryCode = rtcCountryCode;// 可以根据Grs国家码对照表传值，建议传"CN"
cfg.enableHaTrace = YES;
cfg.logLevel = HWRtcLogLevelDebug;//输出DEBUG级别日志
cfg.logPath = [NSString stringWithFormat:@"%@/",logFilePath];//日志存储路径
cfg.enableLog = YES;//开启日志
cfg.logSize = 10*1024;
cfg.muteAudioRoute = NO;//远端音频路由
[_rtcEngine initWithConfig:cfg];
```

步骤2 设置本地窗口。

```
HWRtcVideoCanvas *canvas = [[HWRtcVideoCanvas alloc] init];
canvas.view = [[UIView alloc] initWithFrame:CGRectMacke(0, 0, 90, 160)];//iOS
//canvas.view = [[NSView alloc] initWithFrame:NSMakeRect(0, 0, 90, 160)];//macOS
canvas.uid  = @"HW";
[_rtcEngine setupLocalVideo:canvas viewMode:HWRtcVideoDisplayModeFit];
```

步骤3 加入房间。

```
HWRtcJoinParam *joinParam = [[HWRtcJoinParam alloc]init];
NSString *authorization = @"";// 鉴权信息
joinParam.role  = HWRtcRoleJoiner;
joinParam.userId = @"HW"; // userId用于标识同一房间的不同用户
joinParam.userName = @"HW";// 用户昵称，如无特殊需求，保持和userId一致即可
joinParam.scenario = 1;
joinParam.authorization = authorization;
joinParam.ctime = time;
joinParam.roomId = roomid;
joinParam.autoSubscribeAudio = YES;//是否主动订阅音频
joinParam.autoSubscribeVideo  = NO ;//默认-关闭
BOOL result = [self.rtcEngine joinRoom:joinParam ];
```

joinParam：入会参数，包含用户ID、用户名、房间号、认证信息、ctime、是否自动订阅音频和视频、SFU类型、场景和用户角色，具体请参见**HWRtcJoinParam**。

步骤4 监听远端用户加入房间，并设置远端窗口。

```
-(void)onRemoteUserOnline:(NSString*)roomId userId:(NSString*)userId userName:(NSString*)userName{
    if([userId isEqualToString:localUid]){
        return;
    }
        dispatch_async(dispatch_get_main_queue(),
                       {
            UIView *videoView = [[UIView alloc] initWithFrame:CGRectMake(0, 0, 90, 160)];//iOS
            //NSView *videoView = [[NSView alloc] initWithFrame:NSMakeRect(0, 0, 90, 160)];//macOS
            [self.view addSubview:videoView];
```

```
            HWRtcVideoCanvas *canvas = [[HWRtcVideoCanvas alloc] init];
            canvas.uid  = userId;
            canvas.view = videoView;
            int result = [self.rtcEngine startRemoteStreamView:canvas  treamType:self.streamType];

          if (result == 0) {
              [self.rtcEngine updateRemoteRenderMode:userId displayMode:HWRtcVideoDisplayModeFit
mirrorMode:HWRtcVideoMirrorTypeDisable ];
           }
           [self.viewsArray addObject:canvas];

      });
   }
```

步骤5 监听远端用户离开房间，并删除远端窗口。

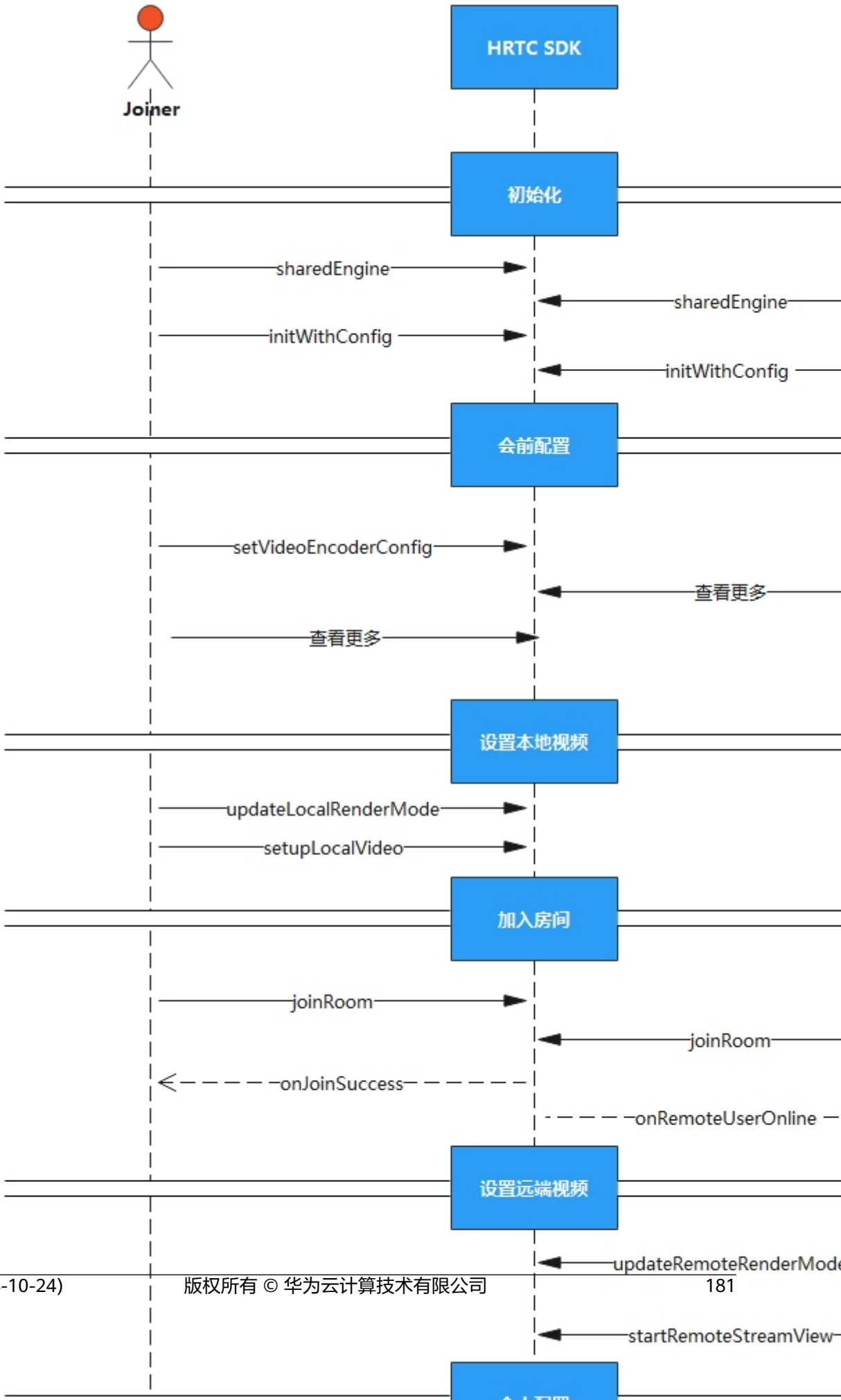
```
- (void)onRemoteUserOffline:(NSString *)roomId userId:(NSString *)userId reason:(NSInteger)reason
{
    dispatch_async(dispatch_get_main_queue(),
    ^{
        for (HWRtcVideoCanvas * canvas in self.viewsArray) {
            if ([userId isEqualToString:canvas.uid]) {
             [self.viewsArray removeObject:canvas];
            }
        }
    });
}
```

步骤6 离开房间。

```
[_rtcEngine leaveRoom];
```

----结束

6.3 基本使用逻辑



1. 创建新的项目工程，导入SDK后，需要先创建引擎。
2. 您可以在入会前进行视频编码、声音播放模式等参数的配置。
3. 设置本地视图。
4. 用户加入房间后，将通过回调的方式通知房间内的其他用户，收到用户加入的回调后，可以为其设置远端视图。
5. 在会中，也可以进行切换摄像头等参数的配置。
6. 离开房间后，需销毁对应资源。

说明

在时序图中，单击相应接口名称可快速跳转到相应接口位置查看其使用方法。

6.4 接口参考

6.4.1 HWRtcEngine

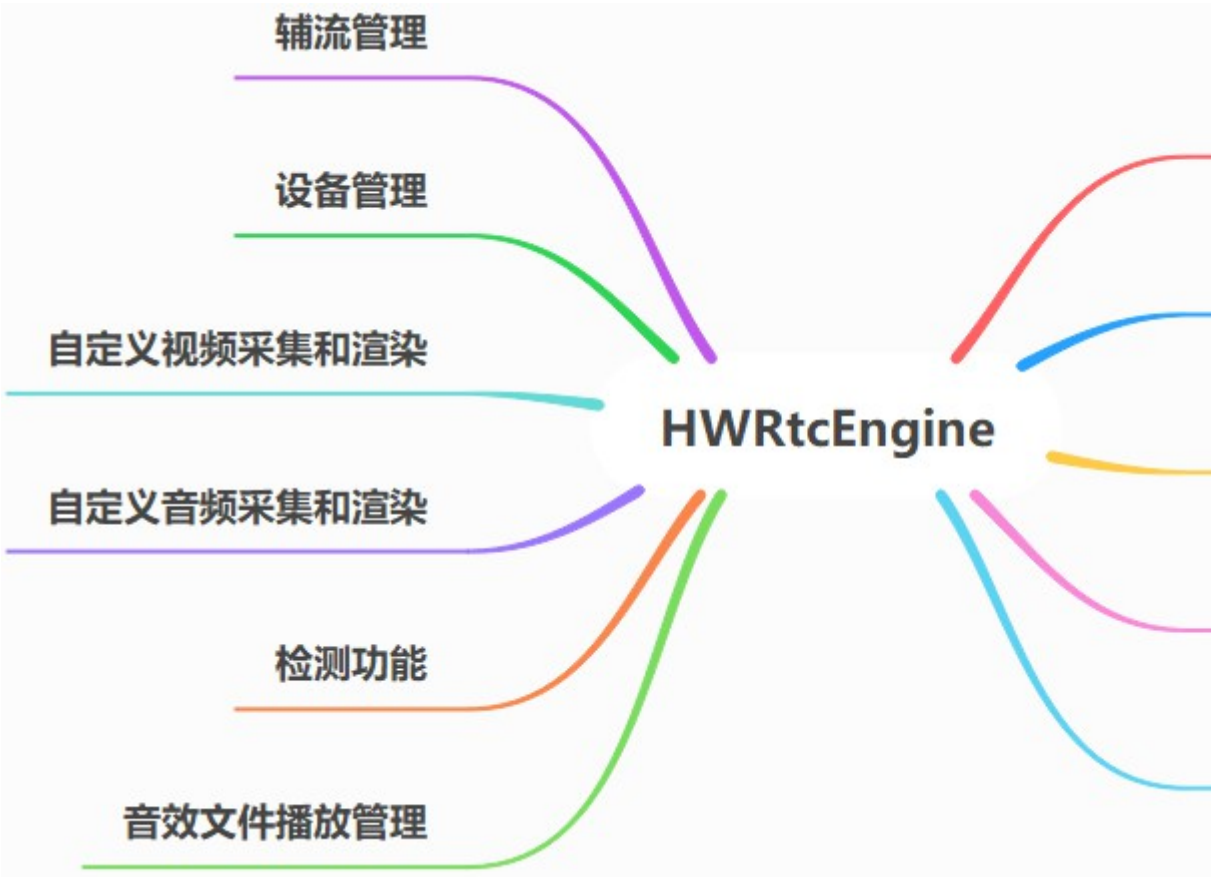
6.4.1.1 接口总览

本章节介绍了iOS/macOS SDK的HWRtcEngine接口详情。

HWRtcEngine按照其功能可分类为：初始化等基础接口、房间功能、视频管理、辅流管理、共享屏幕、音频管理、音效文件播放管理、自定义视频采集和渲染、自定义音频采集和渲染、设备管理、检测功能、音频增强管理。

说明

单击下图中相应接口名称，可快速跳转到相应接口位置查看其使用方法。



初始化等基础接口

表 6-1 初始化等基础接口

接口	描述
sharedEngine	创建连麦引擎实例
destroy	注销引擎
initWithConfig	初始化引擎配置信息
logUpload	上传日志
setEnableRtcStats	打点开关
getVersion	获取SDK版本号
setEncrytionParam	设置端到端加密模式
setAccessResourceType	设置接入的环境
setNetworkBandwidth	设置网络带宽限制

房间功能

表 6-2 房间功能接口

接口	描述
joinRoom	加入房间
changeUserRole	设置用户的角色，切换角色时使用
renewAuthorization	签名更新
leaveRoom	离开房间
changeUserName	设置用户自己的昵称
createConnection	创建 跨房（HWRtcConnection）对象，跨房前需先创建连接
addMultiRoomMediaRelay	添加单个跨房
removeMultiRoomMediaRelay	删除单个跨房
stopMultiRoomMediaRelay	停止所有跨房

音频管理

表 6-3 音频管理接口

接口	描述
muteLocalAudio	设置是否发送本地音频流
enableLocalAudioStream	设置是否开启麦克风采集音频
muteRemoteAudio	订阅或取消订阅远端某个用户音频流
muteAllRemoteAudio	订阅或取消订阅全部远端用户音频流
setVolumeNotifyInterval	设置最大音量上报频率
adjustRecordingVolume	设置麦克风采集的音量（0-100）
adjustPlaybackVolume	调整扬声器播放的音量（0-100）
setShareComputerSound	共享声音 (只支持macOS)
sendAudioSeiMsg	发送音频SEI消息
setDefaultSpeakerModel	设置默认的声音播放模式，在房间外调用
setAudioFrameRecordParameters	设置音频采集回调参数
setAudioConfig	设置音频场景

视频管理

表 6-4 视频管理接口

接口	描述
enableLocalVideo	设置是否开启摄像头采集视频
setVideoEncoderConfig	设置视频编码参数
enableSmallVideoStream	大小流模式设置是否开启小流并设置小流编码参数
startLocalPreview	开启本地预览
stopLocalPreview	关闭本地预览
setupLocalVideo	设置/取消本地预览视图
updateLocalRenderMode	设置本地视频显示模式和镜像模式
setupRemoteView	设置远端流视图
updateRemoteRenderMode	设置远端视图显示模式，镜像模式
setRemoteVideoStreamType	设置选看指定用户的大流或者小流
setPriorRemoteVideoStreamType	设置默认选看远端的大流或者小流
pushLocalVideo	设置是否发送本地视频流
pullRemoteVideo	订阅或取消订阅远端某个用户视频流
pullAllRemoteVideo	订阅或取消订阅全部远端用户视频流
startRemoteStreamView	设置远端用户渲染视图（发起选看-旧接口）
stopRemoteStreamView	关闭远端用户的渲染视图（停止选看）
setRemoteVideoAdjustResolution	设置远端下行视频流的分辨率自适应
setVideoEncoderMirror	设置视频编码镜像模式
startPublishStream	开始旁路推流
updateTransCoding	更新旁路推流
stopPublishStream	停止旁路推流
startAllRemoteView	批量设置远端流视图
appendLocalView	设置本地视频另一个窗口显示
appendRemoteView	设置远端视频另一个窗口显示

辅流管理

表 6-5 辅流管理接口

接口	描述
startRemoteAuxiliaryStreamView	开启辅流渲染视图（发起辅流选看）
stopRemoteAuxiliaryStreamView	关闭辅流渲染视图（停止辅流选看）
updateRemoteAuxiliaryStreamRenderMode	设置辅流视图渲染模式，镜像模式
setRemoteAuxiliaryStreamViewRotation	设置辅流视图角度
setAuxiliaryVideoEncoderConfig	设置辅流编码能力
setAuxiliaryVideoEncodeSmooth	设置是否开启辅流的流畅度优先
pushAuxExternalVideoFrame	推送辅流数据

屏幕共享

表 6-6 屏幕共享接口

接口	描述
startScreenShare	开启屏幕共享
startScreenShareWithAppGroup	根据AppGroup开启屏幕共享（只支持iOS）
stopScreenShare	关闭屏幕共享
getScreenShareSources	获取共享资源列表
setScreenShareTarget	设置共享对象

音效文件播放管理

表 6-7 音频文件播放管理接口

接口	描述
startAudioFile	开始播放音频文件
stopAudioFile	停止播放音频文件
pauseAudioFile	暂停播放音频文件
resumeAudioFile	恢复播放音频文件

接口	描述
adjustAudioFileVolume	调节音频文件播放音量
adjustAudioFilePlayoutVolume	设置本地播放音频音量
getAudioFileVolume	获取音频文件播放音量
getAudioFilePlayoutVolume	获取音频文件本地播放音量
getAudioFileDuration	获取音频文件时长
getAudioFileCurrentPosition	获取当前音频播放位置
setAudioFilePosition	设置音频播放位置
getAudioClipsVolume	获取音效总音量
getVolumeOfAudioClip	获取指定音效音量
setAudioClipsVolume	设置音效总音量
setVolumeOfAudioClip	设置指定音效音量
playAudioClip	播放音效文件
stopAudioClip	停止播放指定音效
stopAllAudioClips	停止播放所有音效
pauseAudioClip	暂停播放指定音效
pauseAllAudioClips	暂停播放所有音效
resumeAudioClip	恢复播放指定音效
resumeAllAudioClips	恢复播放所有音效
getAudioClipCurrentPosition	获取指定音效当前播放位置
setAudioClipPosition	设置指定音效播放位置
getAudioClipDuration	获取音效文件时长

检测功能

表 6-8 检测功能接口

接口	描述
startNetworkTest	会前网络质量开启测试
stopNetworkTest	会前网络质量停止测试

自定义音频采集和渲染

表 6-9 自定义音频采集和渲染接口

接口	描述
setExternalAudioCapture	设置是否开启外部音频采集
pushExternalAudioFrame	输入外部音频数据

自定义视频采集和渲染

表 6-10 自定义视频采集和渲染接口

接口	描述
setExternalVideoCapture	设置是否开启外部视频采集
pushExternalVideoFrame	输入外部视频数据
setExternalMediaFrameOutput	设置媒体数据自渲染
setExternalVideoFrameOutputWithFormat	带Format参数设置媒体数据自渲染
pushAuxExternalVideoFrame	辅流输入外部视频数据

设备管理

表 6-11 设备管理接口

接口	描述
setSpeakerModel	设置声音播放模式（只支持iphone）
setCameraConfig	设置摄像头相关参数
switchCamera	切换前后镜头（只支持iphone）
recordingDeviceTest	音频采集设备测试（只支持macOS）
finishRecordingDeviceTest	结束音频采集设备测试（只支持macOS）
playbackDeviceTest	音频播放设备测试（只支持macOS）
finishPlaybackDeviceTest	结束音频播放设备测试（只支持macOS）
echoTest	音频设备回路测试（只支持macOS）

接口	描述
finishEchoTest	结束音频设备回路测试（只支持macOS）
cameraDeviceTest	视频采集设备测试
finishCameraDeviceTest	结束视频采集设备测试

6.4.1.2 初始化等基础接口

sharedEngine

```
+ (instancetype)sharedEngine;
```

【功能说明】

创建连麦引擎实例。

【请求参数】

无

【返回参数】

返回引擎实例对象。

destroy

```
- (void)destroy;
```

【功能说明】

注销引擎。

【请求参数】

无

【返回参数】

无

initWithConfig

```
- (int)initWithConfig:(HWRtcEngineConfig* _Nonnull)config;
```

【功能说明】

初始化引擎配置信息。

【请求参数】

config：引擎配置，具体请参见HWRtcEngineConfig。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见HWRtcErrorCode。

logUpload

```
- (int)logUpload;
```

【功能说明】

上传日志。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

注意

- 入会成功之后才能主动上传日志。
- 将会触发以下回调：
  - **onLogUploadResult**：日志上传结果回调。
  - **onLogUploadProgress**：日志上传进度回调。

setEnableRtcStats

```
- (int)setEnableRtcStats:(BOOL)enable;
```

【功能说明】

打点开关。

在初始化前或初始化后调用，默认开启。

【请求参数】

enable：YES表示开启，NO表示关闭。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

getVersion

```
+ (NSString* _Nonnull)getVersion;
```

【功能说明】

获取当前SDK版本号。

【返回参数】

SDK当前版本号

setEncrytionParam

```
- (int)setEncrytionParam:(HWRTCEncryptionConfig *_Nonnull)cryptionParam;
```

【功能说明】

设置端到端加密方式。需要在加入房间前设置生效。其中sdk加密模式，需要设置16位加密密钥和加密算法，app加密模式需要先设置回调接口。

【请求参数】

cryptionParam：加密配置，具体请参见**HWRTCEncryptionConfig**。

【返回参数】

- 0：成功。
- <0：失败。具体请参见**HWRtcErrorCode**。

setAccessResourceType

```
- (int)setAccessResourceType:(int)resType;
```

【功能说明】

设置接入的环境，不支持跨房间场景。

【请求参数】

resType：环境类型。

- 0：公网sfu资源。
- 1：公司级sfu。

【返回参数】

- 0：成功。
- <0：失败。具体请参见**HWRtcErrorCode**。

setNetworkBandwidth

```
- (int)setNetworkBandwidth:(HWRtcNetworkBandwidth *)bandwidthParam;
```

【功能说明】

设置网络带宽限制。需要在每次加入房间之前设置。

【请求参数】

bandwidthParam：设置网络带宽限制参数，具体请参见**HWRtcNetworkBandwidth**。

【返回参数】

- 0：成功。
- <0：失败。具体请参见**HWRtcErrorCode**。

6.4.1.3 房间功能

joinRoom

```
- (int)joinRoom:(HWRtcJoinParam * _Nonnull)joinParam;
```

【功能说明】

加入房间。该方法让用户加入通话房间。如果已在通话中，用户必须调用**leaveRoom**退出当前通话，才能进入下一个房间。

【请求参数】

joinParam：用户信息，具体请参见**HWRtcJoinParam**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

会触发以下回调：

- **onConnStateChange**：连接状态发生改变。
- **onJoinSuccess**：加入房间成功。
- **onRemoteUserOnline**：用户收到当前用户加入房间的通知。
- **onJoinRoomFailure**：加入房间失败。

changeUserRole

```
- (int)changeUserRole:(HWRtcRole)role signature:(NSString *)authorization ctime:(long long)ctime;
```

【功能说明】

设置用户在当前房间内的角色类型，角色切换时使用。

【请求参数】

- role：用户角色类型，joiner类型和player类型，具体请参见**HWRtcRole**。
- authorization：预留参数，填null。
- ctime：预留参数，填0。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 加入房间前，可以通过joinRoom的userRole参数确认角色信息。
- 加入指定房间后才可以在指定房间内进行角色切换，当前仅支持joiner和player角色切换。跨房场景下，通过对应connection连接下的changeUserRole接口实现在跨入房间中的角色类型切换。
- 切换成功触发**onUserRoleChange**回调。切换失败会触发**onError**回调，错误码**HWRtcErrorCode** ： HWRtcErrorCodeUserRoleChangeFail。
- 同一时间不同房间最多只能有一个joiner，player切换joiner的时候，需要将其他房间的joiner先切换成player。
- 不支持缺省用户昵称入会。

renewAuthorization

```
- (int)renewAuthorization:(NSString *_Nonnull)signature time:(long long)time;
```

【功能说明】

鉴权签名过期，收到**onAuthorizationExpired**签名鉴权过期回调后，更新鉴权签名。

【请求参数】

- signature：鉴权签名字串。
- time：过期时间。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

leaveRoom

```
- (int)leaveRoom;
```

【功能说明】

离开房间。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 必须调用leaveRoom结束通话后才可以开始下一次通话。
- 会触发以下回调：
  - **onLeaveRoom**：离开房间回调。
  - **onConnStateChange**：连接状态改变回调。
  - **onRemoteUserOffline**：远端用户收到当前用户离开房间的通知。

changeUserName

```
- (int)changeUserName:(NSString *)userName;
```

【功能说明】

房间内设置用户自己的昵称。

【请求参数】

userName：变更的昵称。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 该接口仅支持房间内调用，更改的昵称会被实时同步到房间内其他用户的用户列表，退出房间不会保存，再次加入房间变更为加入房间时设置的昵称（参考**joinRoom**接口注意事项）。
- 会触发用户名变更通知的回调**onUserNameChangedNotify**。

createConnection

```
- (HWRtcConnection *_Nullable)createConnection:(NSString *_Nullable)roomId;
```

【功能说明】

根据房间ID，创建HWRtcConnection对象，为跨房做准备。

通过此接口创建一个与房间关联的HWRtcConnection连接对象。

该方法支持多次调用，创建多个HWRtcConnection连接对象，调用每个对象中的joinRoom方法，可以同时加入到多个房间。在每个房间中，可以分别订阅和选看房间中的用户。

具体请参见**事件回调（HWRtcConnection）**有关接口和回调。

【请求参数】

roomid：房间ID。

【返回参数】

HWRtcConnection：成功返回连接对象指针，失败返回为空。

⚠ 注意

- 同一时间最多只能创建4个连接对象，每个连接对象对应的房间ID必须互不相同。
- 如果使用HWRtcConnection对象加入房间，则加入房间的房间ID不能和已创建连接对象对应的房间ID相同。
- 同一时间只能以JOINER角色加入某一个房间。

addMultiRoomMediaRelay

```
- (int)addMultiRoomMediaRelay:(HWRtcMultiRoomMediaRelayConfiguration
*)roomMediaRelayConfiguration;
```

【功能说明】

添加单个跨房。

【请求参数】

roomMediaRelayConfiguration：跨房信息，具体请参见**HRTCMultiRoomMediaRelayConfiguration**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

removeMultiRoomMediaRelay

```
- (int)removeMultiRoomMediaRelay:(HWRtcMultiRoomMediaRelayConfiguration
*)roomMediaRelayConfiguration;
```

【功能说明】

删除单个跨房。

【请求参数】

roomMediaRelayConfiguration：跨房信息，具体请参见**HRTCMultiRoomMediaRelayConfiguration**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

stopMultiRoomMediaRelay

```
- (int)stopMultiRoomMediaRelay;
```

【功能说明】

停止所有跨房。

【请求参数】

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

6.4.1.4 音频管理

muteLocalAudio

```
- (int)muteLocalAudio:(BOOL)mute;
```

【功能说明】

设置是否关闭本地音频流发送。

【请求参数】

mute：YES表示关闭音频流发送，NO表示开启音频流发送。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

注意

- 默认开启本地音频发流。
- 关闭本地音频发流，不影响本地音频采集。
- 远端用户订阅该用户时，远端用户会收到**onRemoteAudioStateChange**远端流状态变化回调。

enableLocalAudioStream

```
- (int)enableLocalAudioStream:(BOOL)enable;
```

【功能说明】

设置是否开启本地麦克风采集音频。

【请求参数】

enable：YES表示开启，NO表示关闭。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

注意

- 默认开启，关闭不会影响音频流发送。
- 远端用户订阅该用户时，远端用户会收到**onRemoteAudioStateChange**远端流状态变化回调。

muteRemoteAudio

```
- (int)muteRemoteAudio:(NSString *)userid muted:(BOOL)muted;
```

【功能说明】

订阅或取消订阅对应远端用户的音频流。同一时间所有房间最多只能接收17路音频流。

【请求参数】

- userid : 远端用户的id，唯一的标识。
- muted：YES表示取消订阅，NO表示订阅。默认值 NO。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

muteAllRemoteAudio

```
- (int)muteAllRemoteAudio:(BOOL)muted;
```

【功能说明】

订阅或取消订阅当前房间全部用户的音频流。

【请求参数】

muted：YES表示取消订阅，NO表示订阅。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 取消所有音频流接收，同时也会取消接收新加入用户的音频流。
- 开启所有音频流接收，同时也会开启接收新加入用户的音频流。
- 默认开启所有音频流接收。

setVolumeNotifyInterval

```
- (int)setVolumeNotifyInterval:(int)volInterval;
```

【功能说明】

指定音量提示的时间间隔， 设置后按时间间隔触发回调，包括用户音量回调**onUserVolumeStatsNotify**，本地采集音量回调**onLocalVolumeChangedNotify**。

【请求参数】

volInterval：间隔时间, 0 关闭音量上报不再触发回调 。volInterval取值范围为[100 10000]ms。建议设置为2000ms，默认值为2000ms。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

adjustRecordingVolume

```
- (int)adjustRecordingVolume:(unsigned int)volume;
```

【功能说明】

调整录制音量值。

【请求参数】

volume：音量值，取值范围为[0,100]，默认音量值为10， 此接口不会影响系统音量。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败，具体请参见**HWRtcErrorCode**。

adjustPlaybackVolume

```
- (int)adjustPlaybackVolume:(unsigned int)volume;
- (int)adjustPlaybackVolume:(NSString *)userid volume :(unsigned int)volume;
```

【 功能说明 】

设置扬声器播放的音量。

【 请求参数 】

- volume：范围为[0-100]，其中10表示原始音量。
- userid：用户id。

【 返回参数 】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

不影响系统音量。

setShareComputerSound

```
- (int)setShareComputerSound:(BOOL)enable;
```

【 功能说明 】

共享声音。

【 请求参数 】

enable：YES表示开启，NO表示关闭。 默认关闭。

【 返回参数 】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 用户需要**提交工单**联系华为云技术客服获取共享音频驱动。
- 安装驱动有两种方式推荐，任选其一：

  由于第二种方法无法覆盖安装驱动，因此建议您使用第一种集成方式，以对应驱动的版本更新。

  - 在安装集成了SDK的应用时，一同安装驱动。执行如下脚本：
    ```
    sudo installer -pkg "pkg安装包全路径+文件名" -target /Library/Audio/Plug-Ins
    ```
  - 应用启动共享功能时，让SDK自主安装，此时需要做前置动作：将驱动放置在用户的Documents/Resource文件夹（如果Documents目录下没有Resource文件夹，则需要创建），需要用户权限来安装驱动，安装完成后需要重启应用。安装成功后，之后打开共享功能无需再安装。
- SDK无法检测到应用的卸载动作，所以需要应用层面在卸载时删除驱动缓存，执行如下脚本：
  ```
  sudo rm -rf /Library/Audio/Plug-Ins/HAL/HWMeetAudioDevice.driver
  sudo launchctl kickstart -k system/com.apple.audio.coreaudiod
  ```
- 检测驱动安装是否成功：
  1. 查看**/Library/Audio/Plug-Ins/HAL/HWMeetAudioDevice.driver**是否存在。
  2. 查看系统设置偏好-声音-输出一栏，是否切换为hw驱动作为输出。该检查方式只有在启动了共享时才可查看。

sendAudioSeiMsg

```
- (int)sendAudioSeiMsg:(NSString *)message repeateCount:(int)repeateCount;
```

【功能说明】

发送音频SEI消息。通过音频SEI可将自定义信息嵌入到音频流中，发送给其他用户。

【请求参数】

- message：发送的内容。长度为1-500字节。
- repeateCount：发送次数（1-10）。根据需要填发送次数，一般发1次。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

setDefaultSpeakerModel

```
- (int)setDefaultSpeakerModel:(HWRtcSpeakerModel)speakerModel;
```

【功能说明】

设置默认的声音播放模式，在房间外调用。

【请求参数】

speakerModel：声音播放模式，具体请参见**HWRtcSpeakerModel**。默认值为HWRtcSpeakerModelSpeaker。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setAudioFrameRecordParameters

```
- (int)setAudioFrameRecordParameters:(int)sampleRate channel:(int)channel mode:
(HWRTCAudioOperateMode)mode samplesPerCall:(int)samplesPerCall;
```

【功能说明】

设置采集回调参数，配合setAudioFrameObserver的onAudioFrameRecord使用。

【请求参数】

- sampleRate：**onAudioFrameRecord**中返回的采样率，可设置为8000，16000，32000，44100，48000。
- channel：声道，1表示单声道，2表示双声道。
- mode：可读可写模式，具体请参见**HWRTCAudioOperateMode**。
- samplesPerCall：每次回调的单声道样点数（小于(sampleRate/100)*channel*2*3, 大于(sample/(100*3))*channel*2）。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setAudioConfig

```
- (int)setAudioConfig:(HWRtcAudioQualityLevel)level scene:(HWRtcAudioSceneType)scene;
```

【功能说明】

设置使用场景。该接口需要在**joinRoom**前调用。

此接口可在initWithConfig接口设置场景后改变音频场景，暂不支持初始化scene设置音乐再通过此接口设置为会议。

【请求参数】

- level：表示档位，会议模式暂时只支持16k。具体请参见**HWRtcAudioQualityLevel**。
- scene：表示音频模式，具体请参见**HWRtcAudioSceneType**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

6.4.1.5 视频管理

enableLocalVideo

```
- (int)enableLocalVideo:(BOOL)enable;
```

【功能说明】

设置是否开启摄像头采集视频。

【请求参数】

enable：YES表示开启，NO表示关闭。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 默认开启本地视频采集，关闭不会影响视频流发送。
- 远端用户订阅了本端用户时，会收到**onRemoteVideoStateChange**远端流状态变化回调。

setVideoEncoderConfig

```
- (int)setVideoEncoderConfig:(HWRtcVideoEncode * _Nonnull)encParam;
- (int)setVideoEncoderConfig:(NSArray<HWRtcVideoEncode*>*)videoEncodes totalBitRate:(int)totalBitRate;
```

【功能说明】

接口一：大小流模式设置大流发流编码参数。大流必须开启，小流建议开启。

接口二：多流模式，设置发流编码参数。可支持720P到180P的三路流同时推送。

【请求参数】

- encParam：视频编码参数。包括流类型、宽、高、码率、帧率等。具体请参见**HWRtcVideoEncode**。
- videoEncodes：视频编码参数数组，包括流类型、宽、高、码率、帧率等。具体请参见**HWRtcVideoEncode**。
- totalBitRate：总带宽。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 华为SDK系统有默认的编码设置（720P+360P），一般可以不设置发流编码参数。如果您确实需要自定义发流编码参数，请参考华为SDK系统推荐码表：**表6-82**和**表6-83**，否则可能设置失败。
- 两重构接口针对不同使用方式并不通用，如果您使用的新选看系列接口**setupRemoteView**，需配套使用接口一大小流模式设置发流编码参数的**setVideoEncoderConfig**，**enableSmallVideoStream**接口来自定义您的发流编码参数。同理使用旧的选看接口**startRemoteStreamView**需配套使用接口二:多流模式设置发流编码参数setVideoEncoderConfig接口。
- 同一端大小流或多流模式多路流设置的分辨率需保持一致，否则会设置失败。
- 多终端发流和选看的分辨率不一致时，sdk默认自适应并匹配最接近的分辨率（以实际发流分辨率优先），可能会导致选看时设置的分辨率和实际收到的流分辨率不一致。
- 调用接口一设置编码参数的分辨率发生变化时，需要先**enableSmallVideoStream**关闭小流，否则会因为分辨率一致条件限制导致大流设置失败。即涉及分辨率变化时，先关闭小流，再设置大流，再设置小流。
- ios移动端采集帧率限制15帧，建议设置的编码帧率不要超过15帧

enableSmallVideoStream

```
- (int)enableSmallVideoStream:(BOOL)enable encParam:(HWRtcVideoEncode * _Nonnull)encParam;
```

【功能说明】

大小流模式设置是否开启小流，并设置编码参数。小流选择性开启。

【请求参数】

- enable：YES表示开启小流，NO表示关闭小流。
- encParam：视频编码参数。包括流类型、宽、高、码率、帧率等，具体请参见**HWRtcVideoEncode**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

startLocalPreview

```
- (int)startLocalPreview;
```

【功能说明】

开启本地预览。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 该接口限制在房间外调用，在房间内设置不生效，需要调用stopLocalPreview关闭预览，否则将一直处于预览状态。
- 在房间内预览：可调用**setupLocalView**设置有效view开启预览，设置为0表示关闭预览，不需要调用startLocalPreview和stopLocalPreview。
- 在房间外预览：先调用startLocalPreview开启预览，成功后再调用**setupLocalView**设置有效view实现预览；进入房间前需调用stopLocalPreview关闭预览，否则将一直处于预览状态，影响房间内预览。

stopLocalPreview

```
- (int)stopLocalPreview;
```

【功能说明】

关闭本地预览。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

如果调用了**startLocalPreview**，需调用stopLocalPreview关闭预览，否则将一直处于预览状态。该接口限制在房间外调用，在房间内设置不生效。

setupLocalVideo

```
-(int)setupLocalVideo:(HWRtcVideoCanvas *)local viewMode:(HWRtcVideoDisplayMode)viewMode;
- (int)setupLocalVideo:(HWRtcVideoCanvas *)local;
```

【功能说明】

设置/取消本地预览视图。该方法设置本地视频显示信息。 App通过调用此接口绑定本地视频流的显示视窗（view），并设置视频显示模式。

【请求参数】

- local：预览视图，具体请参见**HWRtcVideoCanvas**。
- viewMode：显示模式，具体请参见**HWRtcVideoDisplayMode**。不设置则默认为裁剪模式。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

updateLocalRenderMode

```
-(int)updateLocalRenderMode:(HWRtcVideoDisplayMode)displayMode mirrorMode:
(HWRtcVideoMirrorType)mirrorMode;
```

【功能说明】

设置本地视频显示模式和镜像模式。

【请求参数】

- displayMode：渲染模式。具体请参见**HWRtcVideoDisplayMode**。
- mirrorMode：镜像模式。具体请参见**HWRtcVideoMirrorType**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

setupRemoteView

```
- (int)setupRemoteView:(HWRtcVideoCanvas * _Nonnull)remote;
```

【功能说明】

设置远端流渲染视图（新选看接口）。该接口不影响收流。

【请求参数】

remote: 远端视图，具体请参见**HWRtcVideoCanvas**。remote为nil时表示关闭远端视图。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

updateRemoteRenderMode

```
- (int)updateRemoteRenderMode:(NSString *)userid displayMode:(HWRtcVideoDisplayMode)displayMode
mirrorMode:(HWRtcVideoMirrorType)mirrorMode
```

【功能说明】

设置远端用户视图渲染模式,镜像模式

【请求参数】

- displayMode：视图显示模式，具体请参见**HWRtcVideoDisplayMode**。
- userid：远端用户的唯一标识。
- mirrorMode: 镜像模式，具体请参见HWRtcVideoMirrorType。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

setRemoteVideoStreamType

```
- (int)setRemoteVideoStreamType:(NSString * _Nonnull)userId  type:(HWRtcVideoStreamType)type;
```

【功能说明】

大小流模式，设置选看指定用户的视频流类型。在通过新选看接口发起选看时调用。

【请求参数】

- userId: 远端用户唯一标识。
- type: 视频流类型，指大流、小流，具体请参见**HWRtcVideoStreamType**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

setPriorRemoteVideoStreamType

```
- (int)setPriorRemoteVideoStreamType:(HWRtcVideoStreamType)type;
```

【功能说明】

大小流模式，设置所有订阅的远端视频流类型。默认订阅大流，优先使用**setRemoteVideoStreamType**接口设置的用户流类型。

【请求参数】

type：视频流类型，指大流、小流，具体请参见**HWRtcVideoStreamType**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

pushLocalVideo

```
- (int)pushLocalVideo:(BOOL)push;
```

【功能说明】

设置是否发送本地视频流。

【请求参数】

push：YES表示开启视频流发送，NO表示关闭视频流发送。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 默认开启本地视频发流。
- 关闭本地视频发流，不影响本地视图采集，仍可见本地视图。
- 远端用户订阅了本端用户时，会收到**onRemoteVideoStateChange**远端流状态变化回调。

pullRemoteVideo

```
- (int)pullRemoteVideo:(NSString *)userid pull:(BOOL)pull;
```

【功能说明】

订阅或取消订阅某个用户视频流。只能加入房间后调用。

【请求参数】

- userid：远端用户的id，唯一标识。
- pull：YES表示订阅，NO表示取消订阅。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

pullAllRemoteVideo

```
- (int)pullAllRemoteVideo:(BOOL)pull;
```

【功能说明】

批量订阅或取消订阅当前房间所有远端用户视频流。

【请求参数】

pull：YES表示订阅，NO表示取消订阅，默认开启订阅。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

startRemoteStreamView

```
-(int)startRemoteStreamView:(HWRtcVideoCanvas *)remote  streamType:(HWRtcStreamType)streamType
disableAdjustRes:(BOOL)disableAdjustRes;
```

【功能说明】

设置远端用户渲染视图，并开启收流（选看）。

【请求参数】

- remote：远端预览视图，具体请参见**HWRtcVideoCanvas**。
- streamType：流清晰度，具体请参见**HWRtcStreamType**。
- disableAdjustRes：下行流码率自适应，默认关闭。YES表示关闭，NO表示开启。若关闭，在网络环境较差情况下可能会有卡顿现象。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

该接口为选看的旧接口，通过该接口和updateRemoteRenderMode完成一次完整的选看流程。新的完成选看功能拆分为三个接口：**setupRemoteView**、**pullRemoteVideo**和**setRemoteVideoStreamType**接口，将设置渲染模式、窗口句柄、选看的流类型拆分并增加pullRemoteVideo收流控制接口，以实现更细化的选看流程控制（将窗口绑定和收流控制分开）。您可以根据需要选择调用不同的接口组合以实现视频选看。

stopRemoteStreamView

```
- (int)stopRemoteStreamView:(NSString*)userid;
```

【功能说明】

关闭远端用户的渲染视图，并停止收流（停止选看）。

【请求参数】

userid：远端用户的唯一标识。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

新选看建议通过**setupRemoteView**接口设置窗口为空来停止选看。

setRemoteVideoAdjustResolution

```
- (int)setRemoteVideoAdjustResolution:(BOOL)enable;
```

【功能说明】

设置远端下行视频流分辨率的自适应 。

【请求参数】

enable：YES表示启动自适应，NO表示关闭自适应。默认开启。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

setVideoEncoderMirror

```
- (int)setVideoEncoderMirror:(HWRtcVideoMirrorType)mirror;
```

【功能说明】

设置视频编码（本地发流）画面镜像模式。

【请求参数】

mirror：镜像类型，是否开启镜像。具体请参见**HWRtcVideoMirrorType**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

该接口不改变本地摄像头的预览画面，但会使远端用户看到的和服务器录制的画面为指定的镜像效果。

startPublishStream

```
- (int)startPublishStream:(NSString *)taskId urls:(NSArray *)urls transcodeConfig:(TranscodeConfigModel *)configModel;
```

【功能说明】

开始旁路推流。

【请求参数】

- taskId：任务id，支持自定义，需保证唯一。
- urls：url数组。
- configModel：用户id数组和其他参数，具体请参见**TranscodeConfigModel** 。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

updateTransCoding

```
- (int)updateTransCodingWithTaskId:(NSString *)taskId transcodeConfig:(TranscodeConfigModel *)configModel;
```

【功能说明】

更新旁路推流。在收到远端用户重新入会通知时，需要业务主动调用该接口。

【请求参数】

- taskId：任务id。
- configModel：用户id数组和其他参数，具体请参见**TranscodeConfigModel**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

stopPublishStream

```
- (int)stopPublishStream:(NSString*)taskId;
```

【功能说明】

停止旁路推流。

【请求参数】

taskId：任务id，业务自行定义，保证唯一。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

startAllRemoteView

```
startAllRemoteView(NSArray <HWRtcVideoRemoteView *>*_Nullable)remoteViews count:(unsigned int)count
```

【功能说明】

批量设置远端流视图。

【请求参数】

- remoteViews：必选，数组类型，元素类型为**HWRtcVideoRemoteView**类型。
- count：必选，选看个数，当count为0时，取消所有选看，
  注意：count不能大于remoteViews.count，否则订阅失败。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

appendLocalView

```
- (int)appendLocalView:(HWRtcVideoCanvas *)local
```

【功能说明】

设置本地视频另一个窗口显示。只有在SDK渲染的模式下有效。

【请求参数】

local：窗口句柄。null表示取消扩展的窗口显示，类型为**HWRtcVideoCanvas**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

appendRemoteView

```
- (int)appendRemoteView:(HWRtcVideoCanvas *)remote
```

【功能说明】

设置远程视频另一个窗口显示。只有在SDK渲染的模式下有效。

【请求参数】

remote：窗口句柄。null表示取消扩展的窗口显示，类型为**HWRtcVideoCanvas**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

6.4.1.6 辅流管理

startRemoteAuxiliaryStreamView

```
- (int)startRemoteAuxiliaryStreamView:(HWRtcVideoCanvas *)streamView streamType:
(HWRtcStreamType)streamType;
```

【功能说明】

当远端开启屏幕共享，本地接收到远端屏幕共享开启并通过**onUserAuxiliaryStreamAvailable**回调得到消息后，设置屏幕辅流窗口视图并开始选看（发起辅流选看）。

【请求参数】

- streamView：视图信息，具体请参见**HWRtcVideoCanvas**。
- streamType：编码类型，具体请参见**HWRtcStreamType**。具体设置请参见**表57 不同分辨率下帧率和码率的推荐值**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 收到**onUserAuxiliaryStreamAvailable**消息后，获取对应的userId。
- 多辅流场景，一个用户同时只能订阅一条辅流；当前正在订阅用户A的辅流，需要订阅另一个用户B的辅流时，需要先停止订阅用户A的辅流，再订阅用户B的辅流。

stopRemoteAuxiliaryStreamView

```
- (int)stopRemoteAuxiliaryStreamView:(NSString *)userid;
```

【功能说明】

停止选看辅流视图。

【请求参数】

userid：远端用户的唯一标识。对应**onUserAuxiliaryStreamAvailable**返回的共享用户标识

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

收到**onUserAuxiliaryStreamAvailable**消息后，如果选看的远端屏幕辅流不可用，则必须调用stopRemoteAuxiliaryStreamView关闭。

updateRemoteAuxiliaryStreamRenderMode

```
- (int)updateRemoteAuxiliaryStreamRenderMode:(NSString *)userid
displayMode(HWRtcVideoDisplayMode)displayMode mirrorMode:(HWRtcVideoMirrorType)mirrorMode;
```

【功能说明】

设置辅流视图显示模式。

【请求参数】

- displayMode：显示模式，默认模式为HWRtcVideoDisplayModeFit，具体请参见**HWRtcVideoDisplayMode**。
- userid：远端用户的唯一标识。
- mirrorMode：镜像模式，默认模式为HWRtcVideoMirrorTypeDisable，具体请参见**HWRtcVideoMirrorType**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

setRemoteAuxiliaryStreamViewRotation

```
- (int)setRemoteAuxiliaryStreamViewRotation:(HWRtcVideoRotation)rotation userid:(NSString
*_Nonnull)userid;
```

【功能说明】

设置辅流视图角度。

【请求参数】

- rotation：视图角度，默认角度为HWRtcVideoRotation0，具体请参见**HWRtcVideoRotation**。
- userid：远端用户的唯一标识。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

setAuxiliaryVideoEncoderConfig

```
- (int)setAuxiliaryVideoEncoderConfig:(HWRtcVideoAuxiliarEncParam *_Nonnull)encParam;
```

【功能说明】

设置辅流的编码能力。

【请求参数】

encParam：编码参数设置。具体参考**HWRtcVideoAuxiliarEncParam**；

【返回参数】

- 0：成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

setAuxiliaryVideoEncodeSmooth

```
- (int)setAuxiliaryVideoEncodeSmooth:(BOOL)enable;
```

【功能说明】

设置是否开启辅流的流畅度优先。

【请求参数】

enable：YES表示辅流分辨率为720P；NO表示辅流分辨率为1080P。

【返回参数】

- 0：成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 如果手机分辨率不超过1080p，不可以设置为NO。
- 开启后，辅流发流分辨率为720p，否则发流分辨率为1080p。默认不开启。

pushAuxExternalVideoFrame

```
- (int)pushAuxExternalVideoFrame:(HWRtcVideoFrame * _Nonnull)videoFrame;
```

【功能说明】

推送辅流数据到SDK。

【请求参数】

videoFrame：具体参考**HWRtcVideoFrame**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**

6.4.1.7 屏幕共享

startScreenShare

```
- (int)startScreenShare;
```

【功能说明】

开启屏幕共享功能。

【请求参数】

无

【返回参数】

- 0：成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 目前可支持多路辅流共享，若需开启多辅流，请**提交工单**联系技术支持处理。
- 共享成功后会触发**onScreenShareStarted**回调。
- 开启成功，远端会触发**onUserAuxiliaryStreamAvailable**通知，可据此发起辅流选看。

startScreenShareWithAppGroup

```
- (int)startScreenShareWithAppGroup:(NSString *)appGroup;
```

【功能说明】

开启屏幕共享功能。

【请求参数】

appGroup：NSString 数据类型，需要传入appGroup ID。

【返回参数】

- 0：成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 只支持iPhone。
- 需要先调用**setAuxiliaryVideoEncoderConfig**接口，设置编码参数。
- 目前可支持多路辅流共享，若需开启多辅流，请**提交工单**联系技术支持处理。
- 共享成功后会触发**onScreenShareStarted**回调。
- 开启成功，远端会触发**onUserAuxiliaryStreamAvailable**。

stopScreenShare

```
- (int)stopScreenShare;
```

【功能说明】

关闭屏幕共享功能。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

注意

- 关闭成功，远端会触发onUserAuxiliaryStreamAvailable。
- 关闭成功，本端会触发onScreenRecordFinished。

getScreenShareSources

```
- (NSArray<HWRtcScreenShareSourceInfo *> *)getScreenShareSources;
```

【功能说明】

获取资源共享列表。

【请求参数】

无

【返回参数】

当前可共享的资源列表。具体请参见HWRtcScreenShareSourceInfo。

注意

只支持macOS。

setScreenShareTarget

```
- (int)setScreenShareTarget:(HWRtcScreenShareParam *)param;
```

【功能说明】

设置共享对象。

【请求参数】

param：共享对象的具体信息。具体请参见HWRtcScreenShareParam。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见HWRtcErrorCode。

注意

- 只支持macOS。
- 如果共享对象为桌面类型，则对象名称sourceName不能为空或空字符串，该名称可从getScreenShareSources接口获取。
- 如设置的rect为NSRect(0,0,0,0)，则会根据共享的id和名称取得相应对象的默认区域作为共享范围。

adjustAudioShareVolume

```
- (int)adjustAudioShareVolume:(unsigned int)volume;
```

【功能说明】

设置软增益调整采集的系统声音的音量

【请求参数】

- volume：音量值 [0 , 100]

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setShareComputerScreen

```
- (int)setShareComputerScreen:(BOOL)enable;
```

【功能说明】

设置“声音共享子开关”。当开启媒体共享时，通过该接口可以控制“声音共享”功能的开关。该子开关默认为“关”，并且与 startScreenShare 无调用顺序限制。

【请求参数】

- enable：true表示开启，false表示关闭。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

6.4.1.8 音效文件播放管理

如不集成hwffmpeg.framework，接口则只有startAudioFile/stopAudioFile/pauseAudioFile/resumeAudioFile生效，回调则只有onStartAudioFile/onStopAudioFile/onPauseAudioFile/onResumeAudioFile生效，且startAudioFile播放格式只支持mp3。

startAudioFile

```
- (int)startAudioFile:(HWRtcStartAudioFileParam * _Nonnull)startAudioFileParam;
- (int)startAudioFile:(NSString *)filePath publish:(int)publish cycle:(int)cycle replace:(int)replace;
- (int)startAudioFile:(NSString *)filePath  publish:(int)publish cycle:(int)cycle replace:(int)replace startPos:(unsigned int)startPos;
```

【功能说明】

接口一：设置播放的音频文件，需要入会调用才生效。

接口二：启动音频混音。

接口三：开始播放音频文件。当前只支持本地播放。若角色为“publisher”，不支持调用。

【请求参数】

- startAudioFileParam ： 音频文件参数，具体请参见**HWRtcStartAudioFileParam**。
- filePath：音频文件路径。

- publish：是否将音频发送到远端 ，1表示音频发送到远端，0表示音频仅本地播放。
- cycle：音频播放次数，0表示无限循环。
- replace：是否用音频文件替换麦克风采集的声音，1表示只使用音频文件发送到远端，0表示将本地麦克风采集和音频文件混音后发送到远端。
- startPos：音频文件开始播放位置，单位ms。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

注意

- 需要加入房间后再调用此接口。
- 将会触发**onAudioMixStateChangedNotify**回调。

stopAudioFile

```
- (int)stopAudioFile;
```

【功能说明】

停止播放音频文件。房间内调用。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

注意

- 需要加入房间后再调用此接口。
- 将会触发**onAudioMixStateChangedNotify**回调。

pauseAudioFile

```
- (int)pauseAudioFile;
```

【功能说明】

暂停播放音频文件。房间内调用。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 需要加入房间后再调用此接口。
- 将会触发**onAudioMixStateChangedNotify**回调。

resumeAudioFile

```
- (int)resumeAudioFile;
```

【功能说明】

恢复播放音频文件。房间内调用。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 需要加入房间后再调用此接口。
- 将会触发**onAudioMixStateChangedNotify**回调。

adjustAudioFileVolume

```
- (int)adjustAudioFileVolume:(unsigned int)volume;
```

【功能说明】

调整本地和远端音频播放的音量。

【请求参数】

volume：音量大小，默认音量为100，范围为0-100。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

adjustAudioFilePlayoutVolume

```
- (int)adjustAudioFilePlayoutVolume:(unsigned int)volume;
```

【功能说明】

调整本地音频播放的音量。

【请求参数】

volume：音量大小，默认音量为100，范围为0-100。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

getAudioFileVolume

```
- (int)getAudioFileVolume;
```

【功能说明】

获取音频播放的音量。

【请求参数】

无

【返回参数】

- >=0：音量大小。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

getAudioFilePlayoutVolume

```
- (int)getAudioFilePlayoutVolume;
```

【功能说明】

获取本地播放音频音量。

【请求参数】

无

【返回参数】

音频音量。

注意

需要加入房间后再调用此接口。

getAudioFileDuration

```
- (int)getAudioFileDuration;
```

【功能说明】

获取音频文件的时长。

【请求参数】

无

【返回参数】

- >0：音频时长，单位ms。
- <= 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

需要加入房间后再调用此接口。

getAudioFileCurrentPosition

```
- (int)getAudioFileCurrentPosition;
```

【功能说明】

获取音频文件当前播放位置。

【请求参数】

无

【返回参数】

- >=0：播放位置，单位ms。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

需要加入房间后再调用此接口。

setAudioFilePosition

```
- (int)setAudioFilePosition:(unsigned long long)position;
```

【功能说明】

设置音频播放位置。

【请求参数】

position：播放位置，单位ms。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

需要加入房间后再调用此接口。

getAudioClipsVolume

```
- (int)getAudioClipsVolume;
```

【功能说明】

获取音效总音量。

【请求参数】

无

【返回参数】

- >=0：音量。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

注意

需要加入房间后再调用此接口。

getVolumeOfAudioClip

```
- (int)getVolumeOfAudioClip:(int)soundId;
```

【功能说明】

获取指定音效音量。

【请求参数】

soundId：音效ID。

【返回参数】

- >=0：音量。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

注意

需要加入房间后再调用此接口。

setAudioClipsVolume

```
- (int)setAudioClipsVolume:(double)volume;
```

【功能说明】

设置音效总音量。

【请求参数】

volume：音量，默认音量为100，范围为0-100。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

注意

需要加入房间后再调用此接口。

setVolumeOfAudioClip

```
- (int)setVolumeOfAudioClip:(int)soundId volume:(double)volume;
```

【功能说明】

设置指定音效音量。

【请求参数】

- soundId：音效ID。
- volume：音量，默认音量为100，范围为0-100。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

需要加入房间后再调用此接口。

playAudioClip

```
- (int)playAudioClip:(int)soundId
                filePath:(NSString *)filePath
                loop:(int)loop
                pitch:(double)pitch
                pan:(double)pan
                gain:(double)gain
                publish:(int)publish
                startPos:(int)startPos;
```

【功能说明】

播放音效文件并启动混音，需要在有joiner加入房间后调用。

【请求参数】

- soundId：音效ID，取值>= 0。
- filePath：音效文件路径，支持本地文件和网络文件。
- loop：音效文件播放次数，0为循环播放。
- pitch：音调大小，（**当前不支持**）。
- pan：空间位置，（**当前不支持**）。
- gain：音量大小，取值范围0-100。
- publish：1表示将音效文件混音后发送到远端，0表示本地播放，不发送到远端。
- startPos：起始播放位置，单位ms。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

soundId需要开发者自己生成并维护，保证不同的soundId对应不同的音效播放实例。同时音效播放完毕或者停止播放后，soundId需主动回收，待下一次播放音效时，建议复用被回收的soundId。

stopAudioClip

```
- (int)stopAudioClip:(int)soundId;
```

【功能说明】

停止播放指定音效。

【请求参数】

soundId：音效ID。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

需要加入房间后再调用此接口。

stopAllAudioClips

```
- (int)stopAllAudioClips;
```

【功能说明】

停止播放所有音效。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

需要加入房间后再调用此接口。

pauseAudioClip

```
- (int)pauseAudioClip:(int)soundId;
```

【功能说明】

暂停播放指定音效。

【请求参数】

soundId 音效ID

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

注意

需要加入房间后再调用此接口。

pauseAllAudioClips

```
- (int)pauseAllAudioClips;
```

【功能说明】

暂停播放所有音效。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

注意

需要加入房间后再调用此接口。

resumeAudioClip

```
- (int)resumeAudioClip:(int)soundId;
```

【功能说明】

恢复播放指定音效。

【请求参数】

soundId 音效ID

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

注意

需要加入房间后再调用此接口。

resumeAllAudioClips

```
- (int)resumeAllAudioClips;
```

【功能说明】

恢复播放所有音效。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

需要加入房间后再调用此接口。

getAudioClipCurrentPosition

```
- (int)getAudioClipCurrentPosition:(int)soundId;
```

【功能说明】

获取指定音效当前播放位置。

【请求参数】

soundId：音效ID。

【返回参数】

- >=0：方法调用成功，返回位置，单位为ms。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

需要加入房间后再调用此接口。

setAudioClipPosition

```
- (int)setAudioClipPosition:(int)soundId pos:(int)pos;
```

【功能说明】

设置指定音效播放位置。

【请求参数】

- soundId：音效ID。
- pos：位置，单位为ms。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

需要加入房间后再调用此接口。

getAudioClipDuration

```
- (int)getAudioClipDuration:(NSString *)filePath;
```

【 功能说明 】

获取音效文件时长。

【 请求参数 】

filePath：音效文件路径。

【 返回参数 】

- >0：方法调用成功，单位为ms。
- <= 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

需要加入房间后再调用此接口。

6.4.1.9 检测功能

startNetworkTest

```
- (int)startNetworkTest:(HWRtcNetworkTestConfig *_Nonnull)netWorkTestConfig;
```

【 功能说明 】

会前网络质量开启测试，房间外调用，要等探测结束后才能加入房间。

【 请求参数 】

netWorkTestConfig：检测配置，具体请参见**HWRtcNetworkTestConfig**。

【 返回参数 】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

会触发**onNetworkTestQuality**、**onNetworkTestResult**回调，并通过回调返回具体的网络测试结果，时间需要20-60s。

stopNetworkTest

```
- (int)stopNetworkTest;
```

【功能说明】

会前网络质量停止测试，房间外调用。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

6.4.1.10 自定义音频采集和渲染

setExternalAudioCapture

```
- (int)setExternalAudioCapture:(BOOL)enable sampleRate:(NSUInteger)sampleRate channels:(NSUInteger)channels;
```

【功能说明】

设置是否开启外部音频采集。需要在加入房间前调用。

【请求参数】

- enable：YES表示开启，NO表示关闭。默认为NO。
- sampleRate：音频采样率，支持16k、24k、32k、44.1k、48k采样率。
- channels：音频声道数，1表示单声道，2表示双声道。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 如果使用外部输入音频数据，需要在加入房间后，按照一定时间间隔调用**pushExternalAudioFrame**接口输入音频数据。
- 不支持房间内切换。
- 自采集音频输入规格：
  - 格式：PCM。
  - 采样率：16k/48k 。
  - 声道数：单声道。
  - 位数：16。

pushExternalAudioFrame

```
- (int)pushExternalAudioFrame:(NSData * _Nonnull)audioData size:(NSUInteger)size;
```

【功能说明】

推送外部音频数据。

【请求参数】

- audioData：音频流数据。
- size：音频输入数据大小。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 此方法调用前，需要先调用**setExternalAudioCapture**设置开启外部音频采集。
- 数据输入周期：10ms。
- 音频输入数据大小：10 * sampleRate * channels * 16 / 8 / 1000。

6.4.1.11 自定义视频采集和渲染

setExternalVideoCapture

```
- (int)setExternalVideoCapture:(BOOL)enabled format:(HWRtcVideoImageFormat)format;
```

【功能说明】

设置是否开启外部视频采集。需要在加入房间前调用。

【请求参数】

- enable：YES表示开启，NO表示关闭。默认为NO。
- format：设置外部采集的视频格式，默认为I420（即yuv420P）。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 加入房间前调用，不支持房间内切换。
- 视频支持格式：默认为I420，format可选格式为yuv420p、rgba和texture2d，如果需要外部传入texture2d编码的数据，需要设置format为texture2d，否则传入yuv420p或者rgba。
- 如果使用外部输入视频数据，需要在加入房间后，按照一定时间间隔调用**pushExternalVideoFrame**接口输入视频数据。

pushExternalVideoFrame

```
- (int)pushExternalVideoFrame:(HWRtcVideoFrame* _Nonnull)videoFrame;
```

【功能说明】

推送外部视频数据。

【请求参数】

videoFrame：视频自采集数据格式 ，具体请参见**HWRtcVideoFrame**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 此方法调用前，需要先调用**setExternalVideoCapture**设置开启外部视频采集。
- 数据输入周期：同视频周期，1/帧率。

setExternalMediaFrameOutput

```
- (int)setExternalMediaFrameOutput:(HWRtcMediaType)mediaType remoteEnable:(BOOL)remoteEnable localEnable:(BOOL)localEnable;
```

【功能说明】

设置是否开启视频流自渲染。开启后，回调**onRenderExternalVideoFrame**中会有视频帧数据上报。

【请求参数】

- mediaType：“audio”表示设置音频，“video”表示设置视频 ，具体请参见**HWRtcMediaType**。
- remoteEnable：YES表示开启远端自渲染， NO表示关闭远端自渲染。默认为NO。
- localEnable：YES表示开启本地自渲染，NO表示关闭本地自渲染。默认为NO。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

setExternalVideoFrameOutputWithFormat

```
- (int)setExternalVideoFrameOutputWithFormat:(HwRtcImageBufferFormat)format remoteEnable:(BOOL)remoteEnable localEnable:(BOOL)localEnable;
```

【功能说明】

设置是否开启视频流自渲染，指定Format参数。开启后，回调**onRenderExternalVideoFrame**中会有视频帧数据上报。

【请求参数】

- format ： 指定输出视频内容帧格式，具体请参见**HwRtcImageBufferFormat**。
- remoteEnable：YES表示开启远端自渲染， NO表示关闭远端自渲染。默认为NO。

- localEnable：YES表示开启本地自渲染，NO表示关闭本地自渲染。默认为NO。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

在**onRenderExternalVideoFrame**回调参数**videoFrame.format**中体现设置的**format**参数。

pushAuxExternalVideoFrame

```
- (int)pushAuxExternalVideoFrame:(HWRtcVideoFrame * _Nonnull)dataFrame;
```

【功能说明】

辅流输入外部视频数据。

【请求参数】

videoFrame：视频数据，具体请参考**HWRtcVideoFrame**。

【返回参数】

- 0：成功。
- > 0：失败。

⚠ 注意

- 此接口与屏幕采集功能互斥。
- 如果需要传输texture2d编码流，需要设置setAuxExternalVideoCapture videoEnable参数为true，format设置为texture2d。

6.4.1.12 设备管理

setSpeakerModel

```
- (int)setSpeakerModel:(HWRtcSpeakerModel)speakerModel;
```

【功能说明】

设置声音播放模式。成功加入房间后才能调用。

【请求参数】

speakerModel：声音播放模式，具体请参见**HWRtcSpeakerModel**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

只支持iphone。

setCameraConfig

```
- (int)setCameraConfig:(HWRTCCameraConfig) config;
```

【功能说明】

设置相机的相关参数，如默认使用的摄像头方向

【请求参数】

config：相机的相关参数，具体请参见**HWRTCCameraConfig**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

switchCamera

```
- (int)switchCamera;
```

【功能说明】

切换前后镜头。开启摄像头后调用才生效。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

只支持iphone。

recordingDeviceTest

```
- (int)recordingDeviceTest:(int)intervalInMilliseconds;
```

【功能说明】

音频采集设备测试。

【请求参数】

intervalInMilliseconds：音量回调周期，单位毫秒，范围100到10000。推荐200ms，可以自行调节。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

> **⚠ 注意**
>
> 只支持macOS。

finishRecordingDeviceTest

```
- (int)finishRecordingDeviceTest;
```

【功能说明】

结束音频采集设备测试。

【请求参数】

无。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

> **⚠ 注意**
>
> 只支持macOS。

playbackDeviceTest

```
-(int)playbackDeviceTest:(NSString* _Nonnull)testAudioFilePath;
```

【功能说明】

音频播放设备测试。

【请求参数】

testAudioFilePath：测试音频文件，只支持WAV格式。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

> **⚠ 注意**
>
> 只支持macOS。

finishPlaybackDeviceTest

```
- (int)finishPlaybackDeviceTest;
```


【功能说明】

结束音频播放设备测试。

【请求参数】

无。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

> **⚠ 注意**
>
> 只支持macOS。

echoTest

```
- (int)echoTest:(int)intervalInMilliseconds;
```

【功能说明】

音频设备回路测试。

【请求参数】

intervalInMilliseconds：音量回调周期，单位毫秒，范围100到10000。推荐200ms，客户可以自行调节。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

> **⚠ 注意**
>
> 只支持macOS。

finishEchoTest

```
- (int)finishEchoTest;
```

【功能说明】

结束音频设备回路测试。

【请求参数】

无。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

> **⚠ 注意**
>
> 只支持macOS。

cameraDeviceTest

```
- (int)cameraDeviceTest:(HWRtcVideoCanvas * _Nonnull)local;
```

【 功能说明 】

结束音频设备回路测试。

【 请求参数 】

local：预览视图，具体请参见**HWRtcVideoCanvas**。

【 返回参数 】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

finishCameraDeviceTest

```
- (int)finishCameraDeviceTest;
```

【 功能说明 】

结束视频采集设备测试。

【 请求参数 】

无。

【 返回参数 】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

6.4.2 事件回调（HWRtcEngine）

本章节介绍了iOS/Mac SDK的回调接口HWRtcEngineDelegate的详情。

表 6-12 事件回调说明

接口	描述
onJoinSuccess	成功加入房间回调
onJoinRoomFailure	加入房间失败回调
onRejoinRoomSuccess	重新加入房间成功回调
onLeaveRoom	离开房间回调
onRemoteUserOnline	用户加入房间回调
onRemoteUserOffline	用户离开房间回调

接口	描述
onUserNameChangedNotify	本地用户昵称变化回调
onRemoteUserNameChangedNotify	远端用户昵称变化回调
onFirstRemoteVideoDecoded	引擎收到第一帧远端视频流并解码成功回调
onFirstRemoteVideoDecoded	引擎收到第一帧远端视频流并解码成功回调
onFirstRemoteAuxiliaryStreamDecoded	引擎收到第一帧远端辅流并解码成功回调
onFirstRemoteAuxiliaryStreamDecoded	引擎收到第一帧远端辅流并解码成功回调
onUserRoleChangedNotify	用户角色切换成功回调
onConnectionChangedNotify	连接状态改变回调
onError	错误回调
onWarning	警告回调
onAuthorizationExpired	签名过期回调
onDeviceStateChangedNotify	设备状态改变回调（仅macOS）
onAudioVolumeChanged	音频音量改变回调（仅macOS）
onLocalVolumeChangedNotify	本地音频采集音量
onLocalAudioMutedStatusDetected	本地静音但检测到音频输入回调
onFirstLocalAudioFrameNotify	本端用户音频首帧发送通知
onFirstLocalVideoFrameNotify	本端用户视频首帧发送通知
onAudioRouteChange	音频路由改变回调（仅iOS）
onLogUploadResult	日志上传结果回调
onLogUploadProgress	日志上传进度回调
onRemoteAudioStateChange	远端用户音频流状态改变回调
onRemoteVideoStateChange	远端视频流状态改变
onUserVolumeStatsNotify	音频音量回调
onRenderExternalVideoFrame	渲染视频帧回调
onPlaybackExternalAudioFrame	渲染音频帧回调
onNetworkTestResult	入会前网络检测状态回调
onNetworkTestQuality	入会前网络检测质量状态回调

接口	描述
onVideoStatsNotify	视频流详情，2s触发一次回调
onAudioStatsNotify	音频流详情，2s触发一次回调
onAuxiliaryStreamStatsNotify	辅流流详情，2s触发一次回调
onUserAuxiliaryStreamAvailable	辅流状态回调
onAudioDeviceTestVolumeNotify	本地音频采集音量
onRenderSuccessNotify	用户视频流渲染成功通知回调
onLocalAudioStateChangedNotify	本地音频流状态改变回调
onLocalVideoStateChangedNotify	本地视频流状态改变回调
onNetworkQualityNotify	加入房间后的网络质量状态回调
onLocalVideoStatsNotify	本端视频统计回调
onRemoteVideoStatsNotify	远端视频统计回调
onLocalAudioStatsNotify	本端音频统计回调
onRemoteAudioStatsNotify	远端音频统计回调
onRemoteVideoResolutionChangedNotify	远端视频大小流改变回调
onStatsNotify	当前会话统计回调
onAudioMixStateChangedNotify	混音音频文件状态改变回调
onAudioClipFinished	音效文件播放结束回调
onSeiSendMsgSuccess	发送音频SEI消息成功
onSeiRecvMsg	收到用户的音频SEI消息
onStartPublishStream	开始旁路（RTMP）推流回调
onUpdateTransCoding	更新旁路（RTMP）推流消息
onStopPublishStream	停止旁路（RTMP）推流消息
onStreamPublishStateChange	RTMP推流状态回调
onTopActiveSpeaker	当前音量最大的用户ID
onScreenShareStarted	屏幕共享开启
onScreenShareStopped	屏幕共享关闭

接口	描述
onAudioDeviceTestVolumeNotify	音频设备测试回调
onRemoteMicrophoneStateChanged	远端麦克风设备状态变更通知
onUserNetworkQualityNotify	加入房间后的用户级网络质量状态回调
onMultiRoomMediaRelayStateChanged	跨房状态回调
onMediaAddressNotify	加入房间成功之后sfuip地址回调

onJoinSuccess

```
-(void)onJoinSuccess:(NSString*)roomId userId:(NSString*)userId;
```

【功能说明】

成功加入房间，触发此回调。

【回调参数】

- roomId：新加入的房间ID。
- userId：新加入房间的用户ID。

onJoinRoomFailure

```
- (void)onJoinRoomFailure:(int)errorCode errorMsg:(NSString * _Nonnull)errorMsg;
```

【功能说明】

加入房间失败，触发此回调。

【回调参数】

- errorCode：错误码，具体请参见HRTCErrorCode。
- errorMsg：错误信息描述。

onRejoinRoomSuccess

```
- (void)onRejoinRoomSuccess:(NSString *)roomid userid:(NSString *)userid;
```

【功能说明】

重新加入房间回调。例如网络异常后重连成功加入房间触发。

【回调参数】

- roomid：房间ID。
- userid：用户ID。

onLeaveRoom

```
- (void)onLeaveRoom:(HWRtcLeaveReason)leaveReason statsInfo:(HWRtcStatsInfo *)statsInfo;
```

【功能说明】

离开房间，触发此回调。

【回调参数】

- leaveReason：离开的房间原因，具体请参见**HWRtcLeaveReason**。
- statsInfo：卡顿统计信息，具体请参见**HWRtcStatsInfo**。

⚠ 注意

APP调用**leaveRoom**接口时，会返回HWRtcLeaveReasonUserLeaveRoom，APP可以通过以下任一方式回退到登录界面。

- APP在调用leaveRoom接口时退到登录界面，或者在收到到onLeaveRoom回调，且回调消息不等于HWRtcLeaveReasonUserLeaveRoom时（防止重复操作），退到登录界面。
- APP只在收到onLeaveRoom消息时退到登录界面。

onRemoteUserOnline

```
- (void)onRemoteUserOnline:(NSString*)roomId userId:(NSString*)userId userName:(NSString*)userName;
```

【功能说明】

远端用户加入房间成功，触发此回调。该回调提示有远端用户加入了房间，并返回新加入用户信息。

【回调参数】

- roomId：房间ID。
- userId：远端用户ID。
- userName：远端用户昵称。

onRemoteUserOffline

```
- (void)onRemoteUserOffline:(NSString*)roomId userId:(NSString*)userId reason:(NSInteger)reason;
```

【功能说明】

远端用户离开当前房间，触发此回调。

本端用户离开当前房间，会回调当前房间所有用户offline。

【回调参数】

- roomId：当前房间的房间ID。
- userId：离开房间的远端用户ID。
- reason：远端用户离线原因，预留参数。

onUserNameChangedNotify

```
-(void)onUserNameChangedNotify:(NSString * _Nonnull)oldUserName newUserName:(NSString * _Nonnull)newUserName;
```

【功能说明】

本端用户昵称变化，触发此回调。

【 回调参数 】

- oldUserName：变更前的昵称。
- newUserName：变更后的昵称。

onRemoteUserNameChangedNotify

```
-(void)onRemoteUserNameChangedNotify:(NSString * _Nonnull)roomId userId:(NSString * _Nonnull)userId
userName:(NSString * _Nonnull)userName;
```

【 功能说明 】

远端用户昵称变化，触发此回调。

【 回调参数 】

- roomId：房间ID。
- userId：用户ID。
- userName：昵称。

onFirstRemoteVideoDecoded

```
- (void)onFirstRemoteVideoDecoded:(NSString*)roomId userId:(NSString*)userId width:(int)width height:
(int)height;
```

【 功能说明 】

引擎收到第一帧远端视频流并解码成功回调。

【 回调参数 】

- roomId：视频流对应的房间ID。
- userId：视频流对应的用户ID。
- width：视频流宽度。
- height：视频流高度。

onFirstRemoteVideoDecoded

```
- (void)onFirstRemoteVideoDecoded:(NSString*)roomId userId:(NSString*)userId width:(int)width height:
(int)height elapsed:(NSUInteger)elapsed;
```

【 功能说明 】

引擎收到第一帧远端视频流并解码成功回调。

【 回调参数 】

- roomId：视频流对应的房间ID。
- userId：视频流对应的用户ID。
- width：视频流宽度。
- height：视频流高度。
- elapsed：本端订阅远端用户视频到首帧解码后消耗的时间，单位ms。

onFirstRemoteAuxiliaryStreamDecoded

```
- (void)onFirstRemoteAuxiliaryStreamDecoded:(NSString*)roomId userId:(NSString*)userId width:(int)width
height:(int)height;
```


【功能说明】

引擎收到第一帧远端视频流并解码成功回调。

【回调参数】

- roomId：视频流对应的房间ID。
- userId：视频流对应的用户ID。
- width：视频流宽度。
- height：视频流高度。

onFirstRemoteAuxiliaryStreamDecoded

```
- (void)onFirstRemoteAuxiliaryStreamDecoded:(NSString*)roomId userId:(NSString*)userId width:(int)width height:(int)height elapsed:(NSUInteger)elapsed;
```

【功能说明】

引擎收到第一帧远端视频流并解码成功回调。

【回调参数】

- roomId：视频流对应的房间ID。
- userId：视频流对应的用户ID。
- width：视频流宽度。
- height：视频流高度。
- elapsed：本端订阅远端用户视频到首帧解码后消耗时间，单位ms。

onUserRoleChangedNotify

```
- (void)onUserRoleChangedNotify:(HWRtcRole)oldRole newRole:(HWRtcRole)newRole;
```

【功能说明】

用户房间内角色切换成功，触发此回调。

【回调参数】

- oldRole：切换前的角色，具体请参见**HWRtcRole**。
- newRole：切换成功后的角色，具体请参见**HWRtcRole**。

onConnectionChangedNotify

```
- (void)onConnectionChangedNotify:(HWRtcConnStateType)state reason:(HWRtcConnChangeReason)reason description:(NSString*)description;
```

【功能说明】

网络连接状态发生变化，触发此回调。

【回调参数】

- state：网络连接状态，具体请参见**HWRtcConnStateType**。
- reason：网络连接状态发生变化原因，具体请参见**HWRtcConnChangeReason**。
- description：错误原因描述。

onError

```
- (void)onError:(NSString*)errorCode errorMsg:(NSString*)errorMsg;
```

【功能说明】

发生错误，触发此回调。返回客户端错误码或者服务端错误码。

【回调参数】

- errorCode：错误码，具体请参见**HRTCErrorCode**。
- errorMsg：错误信息描述。

onWarning

```
- (void)onWarning:(int)warningCode warningMsg:(NSString * _Nonnull)warningMsg;
```

【功能说明】

发生错误，触发此回调。返回客户端错误码或者服务端错误码。

【回调参数】

- warningCode：警告码。
- warningMsg：警告信息描述。

onAuthorizationExpired

```
- (void)onAuthorizationExpired;
```

【功能说明】

鉴权签名过期回调，需要app调用**renewAuthorization**更新签名。

onDeviceStateChangedNotify

```
- (void)onDeviceStateChangedNotify:(NSString*)deviceId deviceType:(HWRtcDeviceType)deviceType
deviceState:(HWRtcDeviceState)deviceState;
```

【功能说明】

设备状态发生变化，触发此回调。

【回调参数】

- deviceId：设备ID。
- deviceType：系统设备类型，具体请参见**HWRtcDeviceType**。
- deviceState：系统设备状态，具体请参见**HWRtcDeviceState**。

⚠ 注意

仅支持macOS。

onAudioVolumeChanged

```
- (void)onAudioVolumeChanged:(HWRtcDeviceType)deviceType volume:(unsigned int)volume;
```

【功能说明】

音频设备音量发生变化，触发此回调。

【回调参数】

- deviceType：系统设备类型，具体请参见**HWRtcDeviceType**。
- volume：音量。

⚠ 注意

仅支持macOS。

onLocalVolumeChangedNotify

```
- (void)onLocalVolumeChangedNotify:(int)volume muted:(int)muted;
```

【功能说明】

本地音频采集音量。

【回调参数】

- volume：音量。
- muted：0表示开启本地音频发送，1表示关闭本地音频发送。

onLocalAudioMutedStatusDetected

```
- (void)onLocalAudioMutedStatusDetected;
```

【功能说明】

本地静音但检测到音频输入回调。

【回调参数】

无

⚠ 注意

通过**setVolumeNotifyInterval**开启并设置回调周期，本端禁音后检测到麦克风有输入后定时上报。上报频率和**setVolumeNotifyInterval**的参数大小有关，参考值建议设置成200。

onFirstLocalAudioFrameNotify

```
- (void)onFirstLocalAudioFrameNotify: (NSUInteger)elapsed;
```

【功能说明】

本地音频首帧发送回调。

【回调参数】

elapsed：从入会到本地音频首帧发送所用的时间，单位ms。

onFirstLocalVideoFrameNotify

```
- (void)onFirstLocalVideoFrameNotify:(NSUInteger)elapsed;
```

【功能说明】

本地视频首帧渲染回调。

【回调参数】

elapsed：从开始采集到本地视频首帧渲染所用的时间，单位ms。

onAudioRouteChange

```
- (void)onAudioRouteChange:(HWRtcAudioRoute)audioRoute;
```

【功能说明】

音频路由发生改变，触发此回调。如房间内插拔耳机触发此回调。

【回调参数】

audioRoute：音频路由类型，具体请参见**HWRtcAudioRoute**。

⚠ 注意

仅支持iOS。

onLogUploadResult

```
- (void)onLogUploadResult:(int)result;
```

【功能说明】

日志上传结果回调。

【回调参数】

result：日志上传结果。

- 0：上传成功。
- 1：上传失败。

onLogUploadProgress

```
- (void)onLogUploadProgress:(int)progress;
```

【功能说明】

日志上传进度回调。

【回调参数】

progress：日志上传进度，取值范围0-100。

onRemoteAudioStateChange

```
- (void)onRemoteAudioStateChange:(NSString * _Nonnull)roomid userid:(NSString * _Nonnull)userid state:(HWRtcRemoteAudioState)state reason:(HWRtcRemoteAudioStateReason)reason;
```

【功能说明】

远端用户音频流状态发生改变，会触发此回调。

【回调参数】

- roomid：房间ID。
- userid：远端用户ID。
- state：远端音频流状态，具体请参见**HWRtcRemoteAudioState**。
- reason：远端音频流状态改变的原因，具体请参见**HWRtcRemoteAudioStateReason**。

onRemoteVideoStateChange

```
- (void)onRemoteVideoStateChange:(NSString * _Nonnull)roomid userid:(NSString * _Nonnull)userid state:(HWRtcRemoteVideoState)state reason:(HWRtcRemoteVideoStateReason)reason;
```

【功能说明】

远端用户视频流状态发生改变，会触发此回调。

【回调参数】

- roomid：房间ID。
- userid ： 远端用户ID。
- state：远端视频流状态，具体请参见**HWRtcRemoteVideoState**。
- reason：远端视频流状态改变的原因，具体请参见**HWRtcRemoteVideoStateReason**。

onUserVolumeStatsNotify

```
- (void)onUserVolumeStatsNotify:(NSArray <HWRtcMediaUsersVolumeInfo *> *_Nullable)usersVolumeArray userNumber:(NSUInteger)userNumber totalVolume:(NSUInteger)totalVolume;
```

【功能说明】

用户音量状态回调。通过**setVolumeNotifyInterval**开启并设置回调周期，定时上报。

【回调参数】

- usersVolumeArray：回调发言人音量信息，具体请参见**HWRtcMediaUsersVolumeInfo**。
- userNumber：上报的用户人数，最多上报4人，包含本地用户。
- totalVolume：总音量。

onRenderExternalVideoFrame

```
- (void)onRenderExternalVideoFrame:(NSString *_Nonnull)roomid meidaDirection:(HWRtcMediaDirection)meidaDirection videoFrame:(HWRtcVideoFrame * _Nonnull)videoFrame;
```

【功能说明】

渲染视频帧回调。需要调用**setExternalVideoFrameOutputWithFormat**接口开启视频自渲染，从而触发该回调。

【回调参数】

- roomid：房间ID。
- meidaDirection：数据源（本地/远端数据），媒体方向，具体请参见**HWRTCMediaDirection**。
- videoFrame：视频帧详情，具体请参见**HWRtcVideoFrame**。

onPlaybackExternalAudioFrame

```
- (void)onPlaybackExternalAudioFrame:(NSString *_Nonnull)roomid meidaDirection:
(HWRtcMediaDirection)meidaDirection audioFrame:(HWRtcAudioFrame * _Nonnull)audioFrame;
```

【 功能说明 】

音频自渲染回调。需要调用setExternalMediaFrameOutput接口开启音频自渲染，从而触发该回调。

【 回调参数 】

- roomid ： 房间ID。
- meidaDirection：数据源（本地/远端数据），具体请参见**HWRTCMediaDirection**。
- audioFrame：音频帧详情，具体请参见**HWRtcAudioFrame**。

onNetworkTestResult

```
- (void)onNetworkTestResult:(HWRtcNetworkTestResult * _Nonnull)result;
```

【 功能说明 】

加房间前网络探测结果回调。

【 回调参数 】

HWRtcNetworkTestResult：回调状态主要包括测试成功与否、上行和下行的网络带宽、丢包、延时和抖动，具体请参见**HWRtcNetworkTestResult**。

onNetworkTestQuality

```
- (void)onNetworkTestQuality:(int)level;
```

【 功能说明 】

加房间前网络探测回调。

【 回调参数 】

level：检测网络质量 ，具体请参见**HWRtcNetworkQualityLevel**。

onVideoStatsNotify

```
- (void)onVideoStatsNotify:(NSArray <HWRtcVideoStatsInfo *> * _Nullable)videoStatsArray remoteVideoInfo:
(NSArray <HWRtcVideoStatsInfo *>* _Nullable)remoteVideoStatsInfos;
```

【 功能说明 】

视频流详情，2s触发一次回调。

只有本地一个用户入会时，不回调该方法。

【 回调参数 】

- videoStatsArray：本地视频发流详情，具体请参见**HWRtcVideoStatsInfo**。
- remoteVideoStatsInfo：远端视频收流详情，具体请参见**HWRtcVideoStatsInfo**。

onAudioStatsNotify

```
- (void)onAudioStatsNotify:(NSArray <HWRtcAudioStatsInfo *> * _Nullable)audioStatsArray
remoteAudioInfo:(NSArray <HWRtcAudioStatsInfo *> * _Nullable)remoteAudioStatsInfos;
```

【功能说明】

音频流详情，2s触发一次回调。

【回调参数】

- audioStatsArray：本地音频发流详情，具体请参见**HWRtcAudioStatsInfo**。
- remoteAudioStatsInfos：远端音频收流详情，具体请参见**HWRtcAudioStatsInfo**。

onAuxiliaryStreamStatsNotify

```
- (void)onAuxiliaryStreamStatsNotify:(NSArray <HWRtcVideoStatsInfo *> * _Nullable)subStreamStatsArray remoteVideoInfo:(NSArray <HWRtcVideoStatsInfo *> * _Nullable)remoteVideoStatsInfos;
```

【功能说明】

辅流详情，2s触发一次回调。

【回调参数】

- subStreamStatsArray：本地辅流发流详情，具体请参见**HWRtcVideoStatsInfo**。
- remoteVideoStatsInfos：远端辅流收流详情，具体请参见**HWRtcVideoStatsInfo**。

onUserAuxiliaryStreamAvailable

```
- (void)onUserAuxiliaryStreamAvailable:(NSString * _Nonnull)roomId userId:(NSString * _Nonnull)userId isAvailable:(BOOL)isAvailable;
```

【功能说明】

辅流状态回调。

【回调参数】

- roomId：房间ID。
- userId：远端用户ID。
- isAvailable：YES表示辅流推送中，NO表示辅流停止推送。

onAudioDeviceTestVolumeNotify

```
- (void)onAudioDeviceTestVolumeNotify:(HWRtcAudioDeviceTestVolumeNotify * _Nonnull)result;
```

【功能说明】

本地音频采集音量回调。

【回调参数】

result：本地音频采集音量结果， 具体请参见**HWRtcAudioDeviceTestVolumeNotify**。

onRenderSuccessNotify

```
- (void)onRenderSuccessNotify:(NSString *_Nonnull)userid isAux:(NSUInteger)isAux;
```

【功能说明】

用户视频流渲染成功通知回调。首帧渲染成功、分辨率变化或视频流中断后恢复触发。

【回调参数】

- userId：用户ID。
- isAux：YES表示是辅流，NO表示不是辅流。

onLocalAudioStateChangedNotify

```
-(void)onLocalAudioStateChangedNotify:(HWRtcLocalAudioState)state
                               reason:(HWRtcLocalAudioStateReason)reason;
```

【功能说明】

本地音频流状态变化回调。

【回调参数】

- state：本地音频流状态。具体请参见 **HWRtcLocalAudioState**。
- reason：本地音频流状态变化原因。具体请参见 **HWRtcLocalAudioStateReason**。

onLocalVideoStateChangedNotify

```
-(void)onLocalVideoStateChangedNotify:(HWRtcLocalVideoState)state
                               reason:(HWRtcLocalVideoStateReason)reason;
```

【功能说明】

本地视频流状态变化回调。

【回调参数】

- state：本地视频流状态。具体请参见**HWRtcLocalVideoState**。
- reason：本地视频流状态变化原因。具体请参见 **HWRtcLocalVideoStateReason**。

onNetworkQualityNotify

```
- (void)onNetworkQualityNotify:(NSArray <HWRtcQualityInfo *> * _Nullable)upStreamQualityArray
    downStreamQualityInfo:(NSArray <HWRtcQualityInfo *> * _Nullable)downStreamQualityArray;
```

【功能说明】

房间内客户端基于流级别的网络质量实时上报，默认开启，每2s上报一次，有数据流时才会回调，音频流、视频流分开回调。

【回调参数】

- upStreamQualityArray：上行网络质量状态。具体请参见**HWRtcQualityInfo**。
- downStreamQualityArray：下行网络质量状态。具体请参见 **HWRtcQualityInfo**。

onLocalVideoStatsNotify

```
- (void)onLocalVideoStatsNotify:(NSArray <HWRtcLocalVideoStats *> * _Nullable)localVideoStatsArray;
```

【功能说明】

本端视频流详情，2s触发一次回调。

【回调参数】

localVideoStatsArray：本地视频发流详情参数，具体请参见**HWRtcLocalVideoStats**。

onRemoteVideoStatsNotify

```
- (void)onRemoteVideoStatsNotify:(NSArray <HWRtcRemoteVideoStats *> *
_Nullable)remoteVideoStatsArray;
```

【功能说明】

远端视频流详情，2s触发一次回调。

【回调参数】

remoteVideoStatsArray：远端视频收流详情参数，具体请参见**HWRtcRemoteVideoStats**。

onLocalAudioStatsNotify

```
- (void)onLocalAudioStatsNotify:(NSArray <HWRtcLocalAudioStats *> * _Nullable)localAudioStatsArray;
```

【功能说明】

本端音频流详情，2s触发一次回调。

【回调参数】

localAudioStatsArray：本地音频发流详情，具体请参见**HWRtcLocalAudioStats**。

onRemoteAudioStatsNotify

```
- (void)onRemoteAudioStatsNotify:(NSArray <HWRtcRemoteAudioStats *> *
_Nullable)remoteAudioStatsArray;
```

【功能说明】

远端音频流详情，2s触发一次回调。

【回调参数】

remoteAudioStatsArray：远端音频收流详情，具体请参见**HWRtcRemoteAudioStats**。

onRemoteVideoResolutionChangedNotify

```
- (void)onRemoteVideoResolutionChangedNotify:(NSString *_Nullable)userId width:(NSInteger)width height:
(NSInteger)height;
```

【功能说明】

远端视频大小改变回调。

【回调参数】

- userId：用户ID。
- width：视频流宽度。
- height：视频流高度。

onStatsNotify

```
- (void)onStatsNotify:(NSArray <HRTCOnStats*> * _Nullable)rtcStatsArray;
```

【功能说明】

当前会话统计回调。

回调参数

hrtcOnStats：会话统计信息，具体请参见**HRTCOnStats**。

onAudioMixStateChangedNotify

```
- (void)onAudioMixStateChangedNotify:(HWRtcAudioFileState)state reason:(HWRtcAudioFileReason)reason value:(NSUInteger)value;
```

【功能说明】

混音音频文件播放状态改变回调。

回调参数

- state：音频文件播放状态，具体请参见 **HWRtcAudioFileState**
- reason：音频文件播放状态变化原因，具体请参见 **HWRtcAudioFileReason**
- value：state为HWRtcAudioFileOpenCompleted表示音频文件的时长，单位ms；state为HWRtcAudioFilePositionUpdate表示当前播放的位置，单位ms。其他情况下，value值无意义。

onAudioClipFinished

```
- (void)onAudioClipFinished:(NSInteger)soundId;
```

【功能说明】

音效文件播放结束回调。

回调参数

soundId：音效ID，取值>= 0。

onSeiSendMsgSuccess

```
- (void)onSeiSendMsgSuccess:(NSString * _Nonnull)message;
```

【功能说明】

发送音频SEI消息成功。

回调参数

message：本人发送成功的消息内容。

onSeiRecvMsg

```
- (void)onSeiRecvMsg:(NSString * _Nonnull)userId message:(NSString * _Nonnull)message
```

【功能说明】

收到userId用户的音频SEI消息。

回调参数

- userId：用户uid。
- message：用户发送的消息。

onStartPublishStream

```
- (void)onStartPublishStream:(NSInteger)code taskId:(NSString *_Nonnull)taskId;
```

【功能说明】

开始旁路（RTMP）推流回调。

【回调参数】

- code：错误码，成功为0，失败请参见**HWRtcErrorCode**。
- taskId：任务Id。

onUpdateTransCoding

```
- (void)onUpdateTransCoding:(NSInteger)code taskId:(NSString *_Nonnull)taskId;
```

【功能说明】

更新旁路（RTMP）推流消息。

【回调参数】

- code：错误码，成功为0，失败请参见**HWRtcErrorCode**。
- taskId：任务Id。

onStopPublishStream

```
- (void)onStopPublishStream:(NSInteger)code taskId:(NSString *_Nonnull)taskId;
```

【功能说明】

停止旁路（RTMP）推流消息。

【回调参数】

- code：错误码，成功为0，失败请参见**HWRtcErrorCode**。
- taskId：任务Id。

onStreamPublishStateChange

```
- (void)onStreamPublishStateChange:(NSInteger)code taskId:(NSString *_Nonnull)taskId urlStatus:(NSArray
<HRTCRtmpUrlInfoModel *> * _Nullable)urlList;
```

【功能说明】

RTMP推流状态回调。

【回调参数】

- code：错误码，成功为0，失败请参见**HWRtcErrorCode**。
- taskId：任务Id。
- urlList：推流的url状态，具体请参见**HRTCRtmpUrlInfoModel**。

onTopActiveSpeaker

```
- (void)onTopActiveSpeaker:(NSString *_Nonnull)userId noStream:(BOOL)noStream;
```

【 功能说明 】

声控画面的用户ID变化时，触发此回调。该回调主要用于0号会场场景。

【 回调参数 】

userId：返回当前声控画面的用户ID。

noStream：该用户是否有视频流。

⚠ 注意

0号会场模式下，SDK会持续监测（ 根据一定时间内用户音量大小 ）当前最活跃的用户，如果最活跃用户发生变化，则触发此回调并上报当前最活跃的用户userId。

onScreenShareStarted

```
- (void)onScreenShareStarted;
```

【 功能说明 】

当前状态是系统录屏开启成功回调。只有系统屏幕录制开启成功，才会开启辅流共享。

【 回调参数 】

无

⚠ 注意

只有iPhone才会生效。

onScreenShareStopped

```
- (void)onScreenShareStopped:(int)reason;
```

【 功能说明 】

当前状态是系统录屏完成回调。主动调用stopScreen，不会触发该回调。只有主动关闭系统的屏幕录制，才会回调。当前状态回调之后，不需要再调用stopScreen停止辅流。

【 回调参数 】

无

⚠ 注意

- 只有iPhone才会生效。
- reason参数目前只返回0。

onAudioDeviceTestVolumeNotify

```
- (void)onAudioDeviceTestVolumeNotify:(HWRtcAudioDeviceTestVolumeNotify * _Nonnull)result;
```

【功能说明】

音频设备测试回调。

【回调参数】

result：回调数据，具体请参见**HWRtcAudioDeviceTestVolumeNotify**。

onRemoteMicrophoneStateChanged

```
- (void)onRemoteMicrophoneStateChanged:(NSString * _Nonnull)userId state:(HWRtcRemoteMicState)state;
```

【功能说明】

远端用户麦克风状态变更通知。

【回调参数】

userId：远端用户userId。

state：麦克风设备状态，具体请参见**HWRtcRemoteMicState**。

onUserNetworkQualityNotify

```
- (void)onUserNetworkQualityNotify:(NSString *_Nullable)roomId userId:(NSString *_Nullable)userId
    txQuality:(HWRtcNetworkQualityLevel)txQuality rxQuality:(HWRtcNetworkQualityLevel)rxQuality;
```

【功能说明】

支持用户上报各自与近端SFU间的上下行网络质量，基于用户级，使本地用户能获取同房间内远端用户与其近端SFU间的网络质量。CMD广播时为房间级，将广播给订阅了此主播流的用户或者此主播被选为TOPN用户且观众订阅了该TOPN用户。

【回调参数】

- roomId：用户所在房间号。
- userId: 上报的用户id，0为本地，非0为远端。
- txQuality：该用户的上行网络质量，具体请参见**HWRtcNetworkQualityLevel**。
- rxQuality：该用户的下行网络质量，具体请参见**HWRtcNetworkQualityLevel**。

⚠ 注意

- 此接口不支持跨房场景、WebRTC场景。
- 不支持RTSA。

onMultiRoomMediaRelayStateChanged

```
- (void)onMultiRoomMediaRelayStateChangedWithRoomId:(NSString *)roomId state:
(HWRtcMultiRoomMediaRelayState)state code:(HWRtcMultiRoomMediaRelayStateCode)code
```

【功能说明】

跨房状态回调。

【回调参数】

roomId：跨房房间号。

state：状态类型，具体请参见**HRTCMultiRoomMediaRelayState**。

code：状态的具体原因，具体请参见**HRTCMultiRoomMediaRelayStateCode**。

onMediaAddressNotify

```
- (void)onMediaAddressNotify:(NSString * _Nonnull)mediaIpv4 mediaIpv6:
```

【功能说明】

加入房间成功之后，回调SFU的ipv4和ipv6地址。

【回调参数】

mediaIpv4：sfu ipv4地址。

mediaIpv6：sfu ipv6地址。

6.4.3 HWRtcConnection

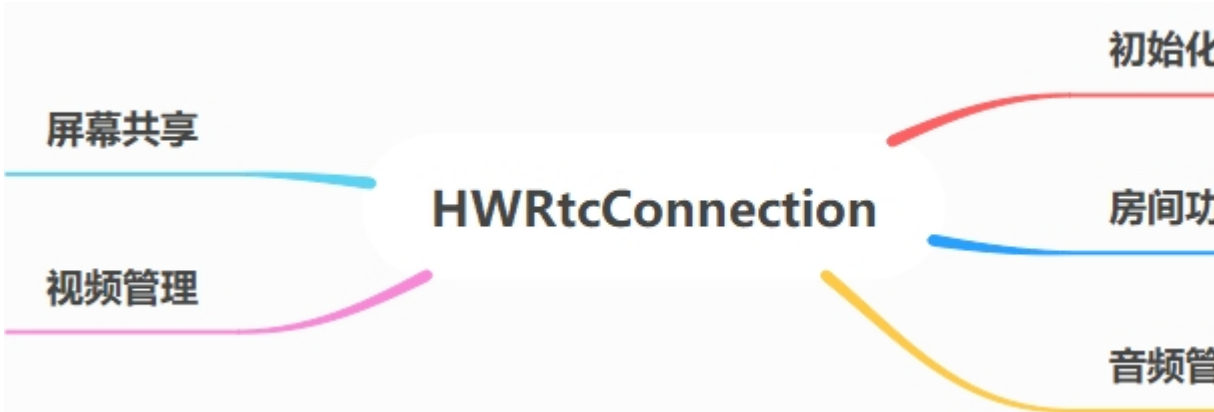
6.4.3.1 接口总览

本章节介绍了iOS/macOS SDK的HWRtcConnection接口详情。

HWRtcConnection按照其功能可分类为：初始化等基础接口、房间功能、视频管理、屏幕共享、音频管理。

说明

单击下图中相应接口名称，可快速跳转到相应接口位置查看其使用方法。



初始化等基础接口

表 6-13 初始化等基础接口

接口	描述
destroyConnection	注销引擎
initWithRoomId	根据房间ID获取connection
setVideoFrameObserver	使能视频前后处理
renewAuthorization	鉴权签名过期，更新签名
setNetworkBandwidth	设置网络带宽限制

房间功能

表 6-14 房间功能接口

接口	描述
joinRoom	加入房间
changeUserRole	设置用户的角色，切换角色时使用
changeUserName	修改用户昵称
getRoomId	获取房间id
leaveRoom	离开房间
disableRejoinRoom	禁止房间自动重入

视频管理

表 6-15 视频管理接口

接口	描述
startRemoteStreamView	按档位选看远端用户，并设置用户窗口
stopRemoteStreamView	取消订阅远端用户，并清除此用户的窗口
updateRemoteRenderMode	设置远端视频渲染填充方式，镜像模式
pullRemoteVideo	订阅或取消订阅远端用户
pullAllRemoteVideo	订阅或取消订阅全部远端用户，对未入会用户也有影响
setupRemoteView	设置远端用户窗口

接口	描述
setRemoteVideoAdjustResolution	远端流自动调整分辨率开关
setPriorRemoteVideoStreamType	大小流模式，设置所有订阅的远端视频流类型
setRemoteVideoStreamType	大小流模式，设置远端视频流类型
setRemoteViewRotation	设置远端视频旋转角度
setRemoteViewOrientation	设置远端视频横屏、竖屏展示

辅流管理

表 6-16 辅流管理接口

接口	描述
startRemoteAuxiliaryStreamView	开启辅流渲染视图（发起辅流选看）
stopRemoteAuxiliaryStreamView	关闭辅流渲染视图（停止辅流选看）
updateRemoteAuxiliaryStreamRenderMode	设置辅流视图渲染模式，镜像模式
setRemoteAuxiliaryStreamViewRotation	设置辅流视图角度
setRemoteAuxiliaryStreamViewOrientation	设置辅流横屏或竖屏显示

音频管理

表 6-17 音频管理接口

接口	描述
muteRemoteAudio	设置是否接收对应远端用户的音频流
muteAllRemoteAudio	设置是否接收所有用户的音频流
adjustPlaybackVolume	调整扬声器播放的音量（0-100）

6.4.3.2 初始化等基础接口

destroyConnection

```
- (void)destroyConnection;
```

【功能说明】

注销引擎。

【请求参数】

无

【返回参数】

无

initWithRoomId

```
- (instancetype)initWithRoomId:(NSString *)roomId;
```

【功能说明】

根据roomId获取connection实例。

【请求参数】

roomId：房间ID。

【返回参数】

返回对应的跨房对象。

setVideoFrameObserver

```
- (int)setVideoFrameObserver:(id <HWRtcConnectionVideoDelegate>)observer;
```

【功能说明】

使能视频前后处理。

【请求参数】

observer：代码对象 要求实现HWRtcConnectionVideoDelegate协议。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

renewAuthorization

```
- (int)renewAuthorization:(NSString *)signature ctime:(long long)ctime;
```

【功能说明】

鉴权签名过期，更新签名。

【请求参数】

- signature：鉴权签名字符串。
- ctime：过期时间。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

setNetworkBandwidth

```
- (int)setNetworkBandwidth:(HWRtcNetworkBandwidth *)bandwidthParam;
```

【功能说明】

设置网络带宽限制。需要在每次加入房间之前设置。

【请求参数】

bandwidthParam：设置网络带宽限制参数，具体请参见**HWRtcNetworkBandwidth**。

【返回参数】

- 0：成功。
- <0：失败。具体请参见**HWRtcErrorCode**。

6.4.3.3 房间功能

joinRoom

```
- (int)joinRoom:(HWRtcJoinParam * _Nonnull)joinParam;
```

【功能说明】

加入房间。该方法让用户加入通话房间。如果已在通话中，用户必须调用**leaveRoom**退出当前通话，才能进入下一个房间。

【请求参数】

joinParam：用户信息，具体请参见**HWRtcJoinParam**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

会触发以下回调：

- **onConnStateChange**：连接状态发送改变。
- **onJoinSuccess**：加入房间成功。
- **onRemoteUserOnline**：远端用户加入。
- **onJoinRoomFailure**：加入房间失败。

changeUserRole

```
- (int)changeUserRole:(HWRtcRole)role signature:(NSString *)authorization ctime:(long long)ctime;
```

【功能说明】

设置用户在本房间内/指定房间的角色，角色切换时使用。

【请求参数】

- role：用户角色，具体请参见**HWRtcRole**。
- authorization：鉴权信息，当填写null时，则切换角色不鉴权，同时会忽略ctime值。
- ctime：生成鉴权时使用的时间戳，必须匹配对应。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 加入房间后才可以切换角色，当前仅支持joiner和player角色切换。
- 切换成功触发**onUserRoleChange**回调。切换失败会触发**onError**回调，错误码**HWRtcErrorCode**：HWRtcErrorCodeUserRoleChangeFail。
- 同一时间不同房间最多只能有一个joiner，player切换joiner的时候，需要将joiner先切换成player，再将当前用户切换成joiner。

changeUserName

```
- (int)changeUserName:(NSString*)userName;
```

【功能说明】

修改用户昵称。

【请求参数】

userName：用户新的昵称。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 该接口仅支持房间内调用，更改的昵称会被实时同步到房间内其他用户的用户列表，退出房间不会保存，再次加入房间变更为加入房间时设置的昵称（参考**joinRoom**接口注意事项）。
- 会触发用户名变更通知的回调onUserNameChangedNotify。

getRoomId

```
- (NSString *)getRoomId;
```

【功能说明】

获取当前房间的roomId。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

disableRejoinRoom

```
- (int)disableRejoinRoom:(BOOL)disable;
```

【功能说明】

禁止自动重入房间。

【请求参数】

disable：使能开关。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

leaveRoom

```
- (int)leaveRoom;
```

【功能说明】

离开房间。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 必须调用leaveRoom结束通话后才可以开始下一次通话。
- 会触发以下回调：
  - **connection:onLeaveRoom**：离开房间回调。
  - **onConnStateChange**：连接状态回调。

6.4.3.4 音频管理

muteRemoteAudio

```
- (int)muteRemoteAudio:(NSString *)userId muted:(BOOL)muted;
```

【功能说明】

设置是否接收对应远端用户的音频流。

【请求参数】

- userId：远端用户的userid，唯一标识。
- muted：YES表示取消订阅，NO表示订阅。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

muteAllRemoteAudio

```
- (int)muteAllRemoteAudio:(BOOL)mute;
```

【功能说明】

设置是否接收所有用户的音频流。

【请求参数】

mute：YES表示取消订阅，NO表示订阅。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 取消所有音频流接收，同时也会取消接收新加入用户的音频流。
- 开启所有音频流接收，同时也会开启接收新加入用户的音频流。
- 默认开启所有音频流接收。

adjustPlaybackVolume

```
- (int)adjustPlaybackVolume:(NSString *)userid volume :(unsigned int)volume;
```

【功能说明】

设置扬声器播放的音量。

【请求参数】

- volume：范围[0-100]，其中10表示原始音量。
- userid：用户id。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

不影响系统音量。

6.4.3.5 视频管理

startRemoteStreamView

```
-(int)startRemoteStreamView:(HWRtcVideoCanvas *)remote  streamType:(HWRtcStreamType)streamType
disableAdjustRes:(BOOL)disableAdjustRes;
```

【功能说明】

设置远端用户窗口，并开启收流。

【请求参数】

- remote：远端预览视图，具体请参见**HWRtcVideoCanvas**。
- streamType：视频分辨率，具体请参见**HWRtcStreamType**。
- disableAdjustRes：禁用分辨率自适应，默认关闭。YES表示关闭，NO表示开启。若关闭，在网络环境较差情况下可能会出现卡顿现象。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

选看当前未选看用户，或者选看用户的流类型发生变化时，会触发回调**onFirstRemoteVideoDecoded**。

stopRemoteStreamView

```
- (int)stopRemoteStreamView:(NSString*)userId;
```

【功能说明】

关闭远端用户的显示视图，并停止收流。

【请求参数】

userId：远端用户的唯一标识。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

updateRemoteRenderMode

```
- (int)updateRemoteRenderMode:(NSString *)userId displayMode:(HWRtcVideoDisplayMode)displayMode
mirrorMode:(HWRtcVideoMirrorType)mirrorMode
```

【功能说明】

设置远端用户视图渲染模式,镜像模式

【请求参数】

- displayMode：视图显示模式，具体请参见**HWRtcVideoDisplayMode**。
- userId：远端用户的唯一标识。

- mirrorMode：镜像模式，具体请参见**HWRtcVideoMirrorType**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

pullRemoteVideo

```
- (int)pullRemoteVideo:(NSString *)userId pull:(BOOL)pull;
```

【功能说明】

开启、关闭指定远端的视频流。

【请求参数】

- userId：远端用户的userid，唯一标识。
- pull：YES表示订阅，NO表示取消订阅。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

pullAllRemoteVideo

```
- (int)pullAllRemoteVideo:(BOOL)pull;
```

【功能说明】

开启、关闭当前所有远端用户的视频流。

【请求参数】

pull：YES表示取消订阅，NO表示订阅。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 取消订阅所有远端用户视频流接收后，同时也会取消接收新加入用户的视频流。
- 开启订阅所有远端用户视频流接收后，同时也会开启接收新加入用户的视频流。
- 默认开启订阅所有远端用户视频流接收。

setupRemoteView

```
- (int)setupRemoteView:(HWRtcVideoCanvas * _Nonnull)view;
```

【功能说明】

设置远端流视图，该接口不影响收流。

【请求参数】

view：远端视图，具体请参见**HWRtcVideoCanvas**。view为nil时，表示关闭远端视图。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

setRemoteVideoAdjustResolution

```
- (int)setRemoteVideoAdjustResolution:(BOOL)enable;
```

【功能说明】

设置是否开启远端流分辨率自适应。默认开启自适应。

【请求参数】

enable：YES表示启动自适应，NO表示关闭自适应。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

setPriorRemoteVideoStreamType

```
- (int)setPriorRemoteVideoStreamType:(HWRtcVideoStreamType)type;
```

【功能说明】

大小流模式，设置所有订阅的远端视频流类型。默认订阅大流，优先应用setRemoteVideoStreamType接口设置的用户流类型。

【请求参数】

type：视频流类型。指大流、小流，具体请参见**HWRtcVideoStreamType**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

setRemoteVideoStreamType

```
- (int)setRemoteVideoStreamType:(NSString *)userId type:(HWRtcVideoStreamType)type;
```

【功能说明】

大小流模式，设置指定选看用户的视频流类型。在通过新选看接口发起选看时调用。

【请求参数】

- userId：远端用户唯一标识。
- type：视频流类型。指大流、小流，具体请参见**HWRtcVideoStreamType**。

【返回参数】

- 0：方法调用成功。

- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

setRemoteViewRotation

```
- (int)setRemoteViewRotation:(NSString *)userId rotation:(HWRtcVideoRotation)rotation;
```

【功能说明】

设置远端流视图的旋转角度。

【请求参数】

- userId：用户ID。
- rotation：旋转角度信息（0°，90°，270°），具体请参见**HWRtcVideoRotation**。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HWRtcErrorCode**。

setRemoteViewOrientation

```
- (int)setRemoteViewOrientation:(NSString *)userId orientation:(HWRtcVideoOrientaion)orientation;
```

【功能说明】

设置远端流视图方向（横竖屏）。

【请求参数】

- userId：用户ID。
- orientation：方向（横竖屏），具体请参见**HWRtcVideoOrientaion**。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HWRtcErrorCode**。

6.4.3.6 辅流管理

startRemoteAuxiliaryStreamView

```
- (int)startRemoteAuxiliaryStreamView:(HWRtcVideoCanvas *)view;
```

【功能说明】

开启辅流渲染视图（发起辅流选看）

【请求参数】

view: 具体请参见HWRtcVideoCanvas

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

- 收到**onUserAuxiliaryStreamAvailable**消息后，获取对应的userId。
- 多辅流场景，一个用户同时只能订阅一条辅流；当前正在订阅用户A的辅流，需要订阅另一个用户B的辅流时，需要先停止订阅用户A的辅流，再订阅用户B的辅流。

stopRemoteAuxiliaryStreamView

```
- (int)stopRemoteAuxiliaryStreamView:(NSString *)userid;
```

【功能说明】

停止选看辅流视图。

【请求参数】

userid：远端用户的唯一标识。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

⚠ 注意

收到**connection:onUserAuxiliaryStreamAvailable**消息后，如果选看的远端屏幕辅流不可用，则必须调用stopRemoteAuxiliaryStreamView关闭。

updateRemoteAuxiliaryStreamRenderMode

```
- (int)updateRemoteAuxiliaryStreamRenderMode:(NSString *)userId
displayMode(HWRtcVideoDisplayMode)displayModemirrorMode:(HWRtcVideoMirrorType)mirrorMode;
```

【功能说明】

设置辅流视图显示模式，镜像模式。

【请求参数】

- displayMode：显示模式，默认模式为HWRtcVideoDisplayModeFit，具体请参见**HWRtcVideoDisplayMode**。
- userId：远端用户的唯一标识。
- mirrorMode：镜像模式，默认模式为HWRtcVideoMirrorTypeDisable，具体请参见**HWRtcVideoMirrorType**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

setRemoteAuxiliaryStreamViewRotation

```
- (int)setRemoteAuxiliaryStreamViewRotation:(NSString *)userId rotation:(HWRtcVideoRotation)rotation;
```

【功能说明】

设置辅流视图角度

【请求参数】

- rotation：视图角度，默认角度为HWRtcVideoRotation0，具体请参见**HWRtcVideoRotation**。
- userId：远端用户的唯一标识。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

setRemoteAuxiliaryStreamViewOrientation

```
- (int)setRemoteAuxiliaryStreamViewOrientation:(NSString *)userId orientation:
(HWRtcVideoOrientaion)orientation;
```

【功能说明】

设置远端辅流视图方向（横竖屏）。

【请求参数】

- userId：用户ID。
- orientation：方向（横竖屏），具体请参见**HWRtcVideoOrientaion**。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HWRtcErrorCode**。

6.4.3.7 媒体原始数据管理

表 6-18 connection 媒体原始数据管理

接口	描述
setVideoFrameObserver	注册原始视频媒体数据监听回调

setVideoFrameObserver

```
- (int)setVideoFrameObserver:(id <HWRtcConnectionVideoDelegate>)observer;
```

【功能说明】

注册原始视频媒体数据监听回调。

【请求参数】

HWRtcConnectionVideoDelegate：原始视频数据处理接口代理。具体请参见**HWRtcConnectionVideoDelegate** 。

【返回参数】

- 0：成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

事件回调（HWRtcConnectionVideoDelegate）

表 6-19 HWRtcConnectionVideoDelegate

接口	描述
connection:onRenderVideoFrame	渲染后视频回调（后处理）

connection:onVideoFrameRender:videoFrame

```
- (BOOL)connection:(HWRtcConnection *)connection
onVideoFrameRender:(NSString *_Nonnull)userid
        videoFrame:(HWRtcVideoFrame* _Nonnull)videoFrame
```

【功能说明】

原始视频数据处理后回调。

【回调参数】

- connection：跨房对象。
- userid：用户ID。
- HWRtcVideoFrame：视频数据格式，具体请参见HWRtcVideoFrame。

【返回参数】

- YES：处理结果成功。
- NO：处理结果失败。

6.4.4 事件回调（HWRtcConnection）

本章节介绍了iOS/macOS SDK的回调接口HWRtcConnectionDelegate的详情。

表 6-20 事件回调说明

接口	描述
connection:onJoinSuccess	成功加入跨房房间回调
connection:onJoinRoomFailure	加入跨房房间失败回调
connection:onRejoinRoomSuccess	重新加入跨房房间成功回调
connection:onLeaveRoom	离开跨房房间回调
connection:onRemoteUserOnline	用户加入跨房房间回调
connection:onRemoteUserOffline	用户离开跨房房间回调
connection:onFirstRemoteVideoDecoded	引擎收到第一帧远端视频流并解码成功回调

接口	描述
connection:onUserRoleChangedNotify	用户角色切换成功回调
connection:onConnectionChangedNotify	连接状态改变回调
connection:onError	错误回调
connectionOnSignatureExpired	签名过期回调
connection:onRemoteAudioStateChange	远端用户音频流状态改变回调
connection:onRemoteVideoStateChange	远端用户视频流状态改变回调
connection:onUserVolumeStatsNotify	音频音量回调
connection:onNetworkQualityNotify	入会中网络质量回调，2s触发一次回调
connection:onVideoStatsNotify	视频流详情，2s触发一次回调
connection:onAudioStatsNotify	音频流详情，2s触发一次回调
connection:onAuxiliaryStreamStatsNotify	辅流流详情，2s触发一次回调
connection:onUserAuxiliaryStreamAvailable	辅流状态回调
connection:onRenderSuccessNotify	用户视频流渲染成功通知回调
connection:onRemoteVideoStatsNotify	远端视频统计回调
connection:onRemoteAudioStatsNotify	远端音频统计回调
connection:onRemoteVideoResolutionChangedNotify	远端视频大小流改变回调
connection:onFirstRemoteAuxiliaryStreamDecoded	远端辅流首帧解码通知
connection:onTopActiveSpeaker	返回当前音量最大的用户ID
connection:onRemoteMicrophoneStateChanged	远端麦克风设备状态变更通知

接口	描述
connection:onMediaAddressNotify	加入房间成功之后sfuip地址回调

connection:onJoinSuccess

```
- (void)connection:(HWRtcConnection *)connection
 onjoinSuccess:(NSString * _Nonnull)userId;
```

【功能说明】

成功加入跨房房间，触发此回调。

【回调参数】

- connection：跨房引擎。
- userId：新加入房间的用户ID。

connection:onJoinRoomFailure

```
- (void)connection:(HWRtcConnection *)connection
  onJoinRoomFailure:(int)errorCode
           errorMsg:(NSString * _Nonnull)errorMsg;
```

【功能说明】

加入房间失败，触发此回调。

【回调参数】

- connection：跨房引擎。
- errorCode：错误码，具体请参见HRTCErrorCode。
- errorMsg：错误信息描述。

connection:onRejoinRoomSuccess

```
- (void)connection:(HWRtcConnection *)connection
 onRejoinRoomSuccess:(NSString * _Nonnull)userid;
```

【功能说明】

重新加入房间回调。例如，网络异常后重连成功加入房间触发。

【回调参数】

- connection：跨房引擎。
- userid：用户ID。

connection:onLeaveRoom

```
-(void)connection:(HWRtcConnection *)connection
       onLeaveRoom:(HWRtcLeaveReason)leaveReason
         statsInfo:(HWRtcStatsInfo * _Nonnull)statsInfo;
```

【功能说明】

离开房间，触发此回调。

【回调参数】

- connection：跨房引擎。
- leaveReason：离开的房间原因，具体请参见**HWRtcLeaveReason**。
- statsInfo：卡顿统计信息，具体请参见**HWRtcStatsInfo**。

⚠ 注意

APP调用**leaveRoom**接口时，会返回HWRtcLeaveReasonUserLeaveRoom，APP可以通过以下任一方式回退到登录界面。

- APP在调用**leaveRoom**接口时退到登录界面，或者在收到onLeaveRoom回调，且回调消息不等于HWRtcLeaveReasonUserLeaveRoom时（防止重复操作），退到登录界面。
- APP只在收到onLeaveRoom消息时退到登录界面。

connection:onRemoteUserOnline

```
- (void)connection:(HWRtcConnection *)connection
     onRemoteUserOnline:(NSString * _Nonnull)userId
       userName:(NSString * _Nonnull)userName;
```

【功能说明】

远端用户加入跨房房间成功，触发此回调。该回调提示有远端用户加入了跨房房间，并返回新加入用户信息。

【回调参数】

- connection：跨房引擎。
- userId：远端用户ID。
- userName：远端用户昵称。

connection:onRemoteUserOffline

```
- (void)connection:(HWRtcConnection *)connection
    onRemoteUserOffline:(NSString * _Nonnull)userId
         reason:(NSInteger)reason;
```

【功能说明】

远端用户离开跨房房间，触发此回调。

本端用户离开当前房间，会回调当前房间所有用户offline。

【回调参数】

- connection：跨房引擎。
- userId：离开房间的远端用户ID。
- reason：远端用户离线原因，预留参数。

connection:onFirstRemoteVideoDecoded

```
- (void)connection:(HWRtcConnection *)connection
 onFirstRemoteVideoDecoded:(NSString * _Nonnull)userId
          width:(int)width height:(int)height;
```

【功能说明】

跨房引擎收到第一帧远端视频流并解码成功回调。

【回调参数】

- connection：跨房引擎。
- userId：用户ID。
- width：视频流宽度。
- height：视频流高度。

connection:onFirstRemoteVideoDecoded

```
- (void)connection:(HWRtcConnection *)connection
 onFirstRemoteVideoDecoded:(NSString * _Nonnull)userId
            width:(int)width height:(int)height elapsed:(NSUInteger)elapsed;
```

【功能说明】

跨房引擎收到第一帧远端视频流并解码成功回调。

【回调参数】

- connection：跨房引擎。
- userId：用户ID。
- width：视频流宽度。
- height：视频流高度。
- elapsed: 首帧解码后消耗的时间，单位ms。

connection:onUserRoleChangedNotify

```
- (void)connection:(HWRtcConnection *)connection
 onUserRoleChangedNotify:(HWRtcRole)oldRole
          newRole:(HWRtcRole)newRole;
```

【功能说明】

用户跨房的房间内角色切换成功，触发此回调。

【回调参数】

- connection：跨房引擎。
- oldRole：切换前的角色，具体请参见**HWRtcRole**。
- newRole：切换成功后的角色，具体请参见**HWRtcRole**。

connection:onConnectionChangedNotify

```
- (void)connection:(HWRtcConnection *)connection
 onConnectionChangedNotify:(HWRtcConnStateType)state
            reason:(HWRtcConnChangeReason)reason
       description:(NSString * _Nonnull)description;
```

【功能说明】

网络连接状态发生变化，触发此回调。

【回调参数】

- connection：跨房引擎。
- state：链接服务器状态，具体请参见**HWRtcConnStateType**。

- reason：连接状态改变原因，具体请参见**HWRtcConnChangeReason**。
- description：连接状态改变描述。

connection:onError

```
- (void)connection:(HWRtcConnection *)connection
          onError:(int)errorCode
         errorMsg:(NSString * _Nonnull)errorMsg;
```

【功能说明】

发生错误，触发此回调。返回客户端错误码或者服务端错误码。

【回调参数】

- connection：跨房引擎。
- errorCode：错误码，具体请参见**HRTCErrorCode**。
- errorMsg：错误信息描述。

connection:onAuthorizationExpired

```
- (void)connectionOnSignatureExpired:(HWRtcConnection *)connection;
```

【功能说明】

鉴权签名过期回调，需要app调用**renewAuthorization**更新签名。

【回调参数】

connection：跨房引擎。

connection:onRemoteAudioStateChange

```
- (void)connection:(HWRtcConnection *)connection
onRemoteAudioStateChange:(NSString * _Nonnull)roomid
            userid:(NSString * _Nonnull)userid
             state:(HWRtcRemoteAudioState)state
            reason:(HWRtcRemoteAudioStateReason)reason;
```

【功能说明】

远端用户音频流状态发生改变，会触发此回调。

【回调参数】

- connection：跨房引擎。
- roomid：房间ID。
- userid：远端用户ID。
- state：远端音频流状态，具体请参见**HWRtcRemoteAudioState**。
- reason：远端音频流状态改变原因，具体请参见**HWRtcRemoteAudioStateReason**。

connection:onRemoteVideoStateChange

```
- (void)connection:(HWRtcConnection *)connection
onRemoteVideoStateChange:(NSString * _Nonnull)roomid
            userid:(NSString * _Nonnull)userid
             state:(HWRtcRemoteVideoState)state
            reason:(HWRtcRemoteVideoStateReason)reason
```


【功能说明】

远端用户视频流状态发生改变，会触发此回调。

【回调参数】

- connection：跨房引擎。
- roomid：房间ID。
- userid：远端用户ID。
- state：远端视频流状态，具体请参见**HWRtcRemoteVideoState**。
- reason：远端视频流状态改变原因，具体请参见**HWRtcRemoteVideoStateReason**。

connection:onUserVolumeStatsNotify

```
- (void)connection:(HWRtcConnection *)connection
 onUserVolumeStatsNotify:(NSArray <HWRtcMediaUsersVolumeInfo *> *_Nullable)usersVolumeArray
      userNumber:(NSUInteger)userNumber totalVolume:(NSUInteger)totalVolume;
```

【功能说明】

用户音量状态回调。通过**setVolumeNotifyInterval**开启并设置回调周期，定时上报。包括总音量以及各发言人及其对应音量上报显示回调

【回调参数】

- connection：跨房引擎。
- usersVolumeArray：回调发言人音量信息，具体请参见**HWRtcMediaUsersVolumeInfo**。
- userNumber：上报的发言人人数，最多上报4人，包含本地用户。
- totalVolume：总音量。

connection:onNetworkQualityNotify

```
- (void)connection:(HWRtcConnection *)connection
  onNetworkQualityNotify:(NSArray <HWRtcQualityInfo *> * _Nullable)upStreamQualityArray
downStreamQualityInfo:(NSArray <HWRtcQualityInfo *> * _Nullable)downStreamQualityArray;
```

【功能说明】

会中基于流级别的网络质量检测回调，音频流、视频流分别回调。

【回调参数】

- connection：跨房引擎。
- upStreamQualityArray：上行网络质量上报。
- downStreamQualityArray：下行网络质量上报。

connection:onVideoStatsNotify

```
- (void)connection:(HWRtcConnection *)connection
     onVideoStatsNotify:(NSArray <HWRtcVideoStatsInfo *> * _Nullable)videoStatsArray
  remoteVideoInfo:(NSArray <HWRtcVideoStatsInfo *>* _Nullable)remoteVideoStatsInfos;
```

【功能说明】

视频流详情，2s触发一次回调。

【回调参数】

- connection：跨房引擎。
- videoStatsArray：本地视频发流详情，具体请参见**HWRtcVideoStatsInfo**。
- remoteVideoStatsInfo：远端视频收流详情，具体请参见**HWRtcVideoStatsInfo**。

connection:onAudioStatsNotify

```
- (void)connection:(HWRtcConnection *)connection
    onAudioStatsNotify:(NSArray <HWRtcAudioStatsInfo *> * _Nullable)audioStatsArray
  remoteAudioInfo:(NSArray <HWRtcAudioStatsInfo *> * _Nullable)remoteAudioStatsInfos;
```

【功能说明】

音频流详情，2s触发一次回调。

【回调参数】

- connection：跨房引擎。
- audioStatsArray：本地音频发流详情，具体请参见**HWRtcAudioStatsInfo**。
- remoteAudioStatsInfos：远端音频收流详情，具体请参见**HWRtcAudioStatsInfo**。

connection:onAuxiliaryStreamStatsNotify

```
- (void)connection:(HWRtcConnection *)connection
 onAuxiliaryStreamStatsNotify:(NSArray <HWRtcVideoStatsInfo *> * _Nullable)subStreamStatsArray
  remoteVideoInfo:(NSArray <HWRtcVideoStatsInfo *> * _Nullable)remoteVideoStatsInfos;
```

【功能说明】

辅流详情，2s触发一次回调。

【回调参数】

- connection：跨房引擎。
- subStreamStatsArray：本地辅流发流详情，具体请参见**HWRtcVideoStatsInfo**。
- remoteVideoStatsInfos：远端辅流收流详情，具体请参见**HWRtcVideoStatsInfo**。

connection:onUserAuxiliaryStreamAvailable

```
- (void)connection:(HWRtcConnection *)connection
onUserAuxiliaryStreamAvailable:(NSString * _Nonnull)userId
      isAvailable:(BOOL)isAvailable;
```

【功能说明】

远端开启，停止辅流后，触发此回调。

【回调参数】

- connection：跨房引擎。
- userId：远端用户ID。
- isAvailable：YES表示远端开启屏幕共享，NO表示远端停止屏幕共享。

connection:onRenderSuccessNotify

```
- (void)connection:(HWRtcConnection *)connection
onRenderSuccessNotify:(NSString *_Nonnull)userId
              isAux:(NSUInteger)isAux;
```

【功能说明】

用户视频流渲染成功通知回调。首帧渲染成功、分辨率变化或视频流中断后恢复触发。

【回调参数】

- connection：跨房引擎。
- userId：用户ID。
- isAux：YES表示是辅流，NO表示不是辅流。

connection:onRemoteVideoStatsNotify

```
- (void)connection:(HWRtcConnection *)connection
onRemoteVideoStatsNotify:(NSArray <HWRtcRemoteVideoStats *> * _Nullable)remoteVideoStatsArray;
```

【功能说明】

远端视频流详情，2s触发一次回调。

【回调参数】

- connection：跨房引擎。
- remoteVideoStatsArray：远端视频收流详情参数，具体请参见**HWRtcRemoteVideoStats**。

connection:onRemoteAudioStatsNotify

```
- (void)connection:(HWRtcConnection *)connection
onRemoteAudioStatsNotify:(NSArray <HWRtcRemoteAudioStats *> * _Nullable)remoteAudioStatsArray;
```

【功能说明】

远端音频流详情，2s触发一次回调。

【回调参数】

- connection：跨房引擎。
- remoteAudioStatsArray：远端音频收流详情，具体请参见**HWRtcRemoteAudioStats**。

connection:onRemoteVideoResolutionChangedNotify

```
- (void)connection:(HWRtcConnection *)connection onRemoteVideoResolutionChangedNotify:(NSString
*_Nullable)userId width:(NSInteger)width height:(NSInteger)height;
```

【功能说明】

远端视频大小改变回调。

【回调参数】

- connection：跨房引擎。
- userId：用户ID。

- width：视频流宽度。
- height：视频流高度。

connection:onFirstRemoteAuxiliaryStreamDecoded

```
-(void)connection:(HWRtcConnection *)connection
onFirstRemoteAuxiliaryStreamDecoded:(NSString * _Nonnull)userId
                              width:(int)width
                              height:(int)height;
```

【功能说明】

引擎收到第一帧远端视频流并解码成功回调。

【回调参数】

- connection：跨房引擎。
- userId：视频流对应的用户ID。
- width：视频流宽度。
- height：视频流高度。

connection:onFirstRemoteAuxiliaryStreamDecoded

```
-(void)connection:(HWRtcConnection *)connection
onFirstRemoteAuxiliaryStreamDecoded:(NSString * _Nonnull)userId
                              width:(int)width
                              height:(int)height
                              elapsed:(NSUInteger)elapsed;
```

【功能说明】

引擎收到第一帧远端视频流并解码成功回调。

【回调参数】

- connection：跨房引擎。
- userId：视频流对应的用户ID。
- width：视频流宽度。
- height：视频流高度。
- elapsed：首帧解码后消耗的时间，单位ms。

connection:onTopActiveSpeaker

```
- (void)connection:(HWRtcConnection *)connection onTopActiveSpeaker:(NSString *_Nonnull)userId
noStream:(BOOL)noStream;
```

【功能说明】

返回当前音量最大的用户ID。

【回调参数】

- connection：跨房引擎
- userId：用户ID。
- noStream：该用户是否有视频流。

connection:onRemoteMicrophoneStateChanged

```
- (void)connection:(HWRtcConnection *)connection
  onRemoteMicrophoneStateChanged:(NSString * _Nonnull)userId state:(HWRtcRemoteMicState)state;
```

【功能说明】

远端用户麦克风状态变更通知。

【回调参数】

- connection：跨房引擎
- userId：远端用户userId。
- state：麦克风设备状态，具体请参见**HWRtcRemoteMicState**。

connection:onMediaAddressNotify

```
- (void)connection:(HWRtcConnection *)connection onMediaAddressNotify:(NSString * _Nonnull)mediaIpv4
mediaIpv6:(NSString * _Nonnull)mediaIpv6;
```

【功能说明】

加入房间成功之后，回调SFU的ipv4和ipv6地址。

【回调参数】

- connection：跨房引擎
- mediaIpv4：sfu ipv4地址。
- mediaIpv6：sfu ipv6地址。

6.4.5 媒体原始数据管理

6.4.5.1 注册回调（IHRTCMediaEngine）

表 6-21 HWRtcMediaEngine

接口	描述
setVideoFrameObserver	注册原始视频媒体数据监听回调
setAudioFrameObserver	注册原始音频媒体数据监听回调

setVideoFrameObserver

```
- (int)setVideoFrameObserver:(id<HWRtcMediaEngineVideoDelegate>)observer;
```

【功能说明】

注册原始视频媒体数据监听回调。

【请求参数】

HWRtcMediaEngineVideoDelegate：原始视频数据处理接口代理。具体请参见**HWRtcMediaEngineVideoDelegate**。

【返回参数】

- 0：成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

setAudioFrameObserver

```
- (int)setAudioFrameObserver:(id<HWRtcMediaEngineAudioDelegate>)observer;
```

【功能说明】

注册原始音频媒体数据监听回调。

【请求参数】

HWRtcMediaEngineAudioDelegate：原始音频数据处理接口代理。具体请参见**HWRtcMediaEngineVideoDelegate**。

【返回参数】

- 0：成功。
- > 0：方法调用失败。具体请参见**HWRtcErrorCode**。

6.4.5.2 事件回调（HWRtcMediaEngineVideoDelegate）

本章节介绍了iOS SDK的回调接口HWRtcMediaEngineVideoDelegate的详情。

表 6-22 事件回调说明

接口	描述
onVideoFrameCapture	原始视频回调（前处理）
onVideoFrameRender	渲染后视频回调（后处理）
requireCaptureVideoFrame	是否开启前处理
requireRenderVideoFrame	是否开启后处理

onVideoFrameCapture

```
- (BOOL)onVideoFrameCapture:(HWRtcVideoFrame* _Nonnull)videoFrame;
```

【功能说明】

原始视频回调，从接口回调中取到原始视频数据以作前处理。

【回调参数】

HWRtcVideoFrame：视频数据格式，具体请参见**HWRtcVideoFrame**。

【返回参数】

- YES：处理结果成功。
- NO：处理结果失败。

onVideoFrameRender

```
- (BOOL)onVideoFrameRender:(NSString *_Nonnull)userid
                videoFrame:(HWRtcVideoFrame* _Nonnull)videoFrame;
```

【功能说明】

原始视频数据处理后回调。

【回调参数】

userid：用户ID

HWRtcVideoFrame：视频数据格式，具体请参见**HWRtcVideoFrame**。

【返回参数】

- YES：处理结果成功。
- NO：处理结果失败。

requireCaptureVideoFrame

```
- (BOOL)requireCaptureVideoFrame;
```

【功能说明】

是否需要开启前处理。

【返回参数】

- YES：开启。
- NO：不开启。

requireRenderVideoFrame

```
- (BOOL)requireRenderVideoFrame;
```

【功能说明】

是否需要开启后处理。

【返回参数】

- YES：开启。
- NO：不开启。

6.4.5.3 事件回调（HWRtcMediaEngineAudioDelegate）

本章节介绍了iOS SDK的回调接口HWRtcMediaEngineAudioDelegate的详情。

表 6-23 事件回调说明

接口	描述
onAudioFramePlayback	音频播放回调（后处理）
onAudioFrameMixed	音频混音处理回调
onAudioFrameRecord	音频采集回调（前处理）
requireRecordAudioFrame	是否开启音频前处理
requirePlaybackAudioFrame	是否开启音频后处理
requireMixedAudioFrame	是否开启音频混音回调

onAudioFramePlayback

```
- (BOOL)onAudioFramePlayback:(HWRtcAudioFrame * _Nonnull)audioFrame;
```

【功能说明】

需要播放的音频数据回调，从接口回调中取到音频数据以作后处理。

【回调参数】

HWRtcAudioFrame ： 音频数据格式，具体请参见**HWRtcAudioFrame**。

【返回参数】

- YES：处理结果成功。
- NO：处理结果失败。

onAudioFrameMixed

```
- (BOOL)onAudioFrameMixed:(HWRtcAudioFrame * _Nonnull)audioFrame;
```

【功能说明】

全部音频混音数据回调，包含上下行所有通道。

【回调参数】

HWRtcAudioFrame：音频数据格式，具体请参见**HWRtcAudioFrame**。

【返回参数】

- YES：处理结果成功。
- NO：处理结果失败。

onAudioFrameRecord

```
- (BOOL)onAudioFrameRecord:(HWRtcAudioFrame * _Nonnull)audioFrame;
```

【功能说明】

音频采集原始数据回调，对音频数据的修改会发送到远端。

【回调参数】

HWRtcAudioFrame：音频数据格式，具体请参见**HWRtcAudioFrame**。

【返回参数】

- YES：处理结果成功。
- NO：处理结果失败。

requireRecordAudioFrame

```
- (BOOL)requireRecordAudioFrame;
```

【功能说明】

是否开启音频前处理。

【返回参数】

- YES：开启。
- NO：不开启。

requirePlaybackAudioFrame

```
- (BOOL)requirePlaybackAudioFrame;
```

【功能说明】

是否需要开启音频后处理。

【返回参数】

- YES：开启。
- NO：不开启。

requireMixedAudioFrame

```
- (BOOL)requireMixedAudioFrame;
```

【功能说明】

是否需要开启全部音频混音数据回调。

【返回参数】

- YES：开启。
- NO：不开启。

6.4.6 HWRtcReplay

表 6-24 HWRtcReplay 接口

接口	描述
sharedInstance	生成HWRtcReplay单例对象
setupWithAppGroup	通过appGroup启动HWrtcReplay
broadcastFinished	系统录屏结束
sendVideoSampleBuffer	发送录屏数据

sharedInstance

```
+ (instancetype)sharedInstance;
```

【功能说明】

生成HWRtcReplay单例对象。

【请求参数】

无。

【返回参数】

无

setupWithAppGroup

```
- (void)setupWithAppGroup:(NSString *)appGroup delegate:(id<HWRtcReplayDelegate>)delegate;
```

【 功能说明 】

通过appGroup启动HWrtcReplay。在sampleHandle类中的broadcastStartedWithSetupInfo:接口内调用。

【 请求参数 】

- appGroup ： appGroupID。
- delegate：回调对象，具体请参见HWRtcReplayDelegate。

【 返回参数 】

无

broadcastFinished

```
- (void)broadcastFinished;
```

【 功能说明 】

系统录屏完成，在sampleHandle类中的broadcastFinished中调用。

【 请求参数 】

无

【 返回参数 】

无

sendVideoSampleBuffer

```
- (void)sendVideoSampleBuffer:(CMSampleBufferRef)sampleBuffer;
```

【 功能说明 】

发送录屏数据。需要在sampleHandle的实现类中的processSampleBuffer: 方法中调用。

【 请求参数 】

sampleBuffer：录屏数据。

【 返回参数 】

无

6.4.7 客户端错误码

本章节介绍了iOS/macOS SDK的客户端错误码HWRtcErrorCode的详细信息。

表 6-25 错误码说明

类成员	错误码	描述	错误原因
HWRtcErrorCodeSuccess	0	调用成功	-
HWRtcErrorCodeSdkInternalError	90000001	sdk内部系统错误	SDK内部异常。
HWRtcErrorCodeMsgToolarge	90000002	发送的消息太大	发送消息时，消息体太大。
HWRtcErrorCodeMemNotEnough	90000003	内存不足	内存无法申请。
HWRtcErrorCodeSynsendMsgError	90000004	消息发送失败	消息队列异常，导致内部消息发送失败。
HWRtcErrorCodeParamError	90000005	参数错误	包括如下两方面： • 接口入参无效。 • 内部参数错误。
HWRtcErrorCodeApiCalledInWrongOrder	90000006	api接口调用顺序不当	当前只有日志设置必须在初始化之前。
HWRtcErrorCodeSetupLocalViewFail	90000007	设置本地窗口失败	player场景没有本地画面，不应该设置。
HWRtcErrorCodeSetupRemoteViewFail	90000008	设置远端窗口失败	publisher场景没有远端画面，不应该设置。
HWRtcErrorCodeSetDeviceFail	90000009	设置设备失败	设置播放、录音、视频设备失败。
HWRtcErrorCodeInitializing	90000010	初始化过程中	初始化过程中，不能再做初始化或者去初始化操作。
HWRtcErrorCodeUnInitializing	90000011	去初始化过程中	去初始化过程中，不能再做初始化或者去初始化操作。
HWRtcErrorCodeLogUploading	90000012	日志正在上传	日志正在上传过程中。
HWRtcErrorCodeMediaPortError	90000013	媒体端口获取失败	音频从10010开始，视频从10020开始，尝试10次，端口被占用。
HWRtcErrorCodeWatchViewTooMuch	90000014	选看超过规格	当前支持最多设置16个设置远端窗口，若超过，则会失败。
HWRtcErrorCodeMediaCmpErr	90000015	媒体协商失败	与服务器之间媒体协商失败。
HWRtcErrorCodeServerNoResponse	90000016	服务器无响应	服务器无响应。

类成员	错误码	描述	错误原因
HWRtcErrorCodeUserRoleChangeFail	90000017	切换角色失败	切换角色失败。
HWRtcErrorCodeJoinRoomFail	90000018	加入房间失败	加入房间失败。
HWRtcErrorCodeJoinRoomStatusBusy	90000019	会议状态不是空闲态	已在房间中或正在进行网络探测。
HWRtcErrorCodejoinRoomServerError	90000020	加入房间服务器错误	加入房间服务器错误。
HWRtcErrorCodejoinRoomServiceUnreachable	90000021	加入房间服务器无法访问	加入房间服务器无法访问。
HWRtcErrorCodejoinRoomAuthFail,	90000022	加入房间验证错误	加入房间验证错误。
HWRtcErrorCodejoinRoomAuthRetry	90000023	加入房间失败	加入房间失败，鉴权重试。
HWRtcErrorCodejoinRoomClockSync	90000024	加入房间失败	加入房间失败，时钟同步。
HWRtcErrorCodejoinRoomUrlNotRight	90000025	加入房间失败	加入房间失败，url错误
HWRtcErrorCodeKickedOff	90000026	被踢出房间	相同用户ID等原因，被踢出房间。
HWRtcErrorCodeScreenShareFail	90000027	共享失败	共享失败。
HWRtcErrorCodeExtMediaOutPut	90000028	当前为外部媒体输出模式，禁用该操作	当前为外部媒体输出模式，禁用该操作。
HWRtcErrorCodeReconnectFail	90000029	重连失败	重连失败。
HWRtcErrorCodeBreakDown	90000030	服务器宕机	服务器宕机。
HWRtcErrorCodeSignatureExpired	90000031	签名已经过期	签名已经过期。
HWRtcErrorCodeSetRemoteRenderModeFail	90000032	设置远端窗口模式失败	设置远端窗口模式失败。
HWRtcErrorCodeSetRemoteAudioMuteFail	90000033	订阅或取消订阅音频失败	订阅或取消订阅音频失败。
HWRtcErrorCodeConnectOtherRoomFail	90000034	连接其他房间失败	连接其他房间失败。

类成员	错误码	描述	错误原因
HWRtcErrorCodeDisConnectOtherRoomFail	90000035	断开连接其他房间失败	断开连接其他房间失败。
HWRtcErrorCodeSetUseRoleNotAllowed	90000036	不允许角色切换	不允许角色切换。
HWRtcErrorCodeExtMediaCaptureInput	90000037	当前为第三方采集模式，禁用该操作	当前为第三方采集模式，禁用该操作。
HWRtcErrorCodeSetExtAudioCaptureFail	90000038	设置第三方音频采集失败	设置第三方音频采集失败。
HWRtcErrorCodeSetExtVideoCaptureFail	90000039	设置第三方视频采集失败	设置第三方视频采集失败。
HWRtcErrorCodeSetShareComputerSoundFail	90000040	设置共享声音开关失败	设置共享声音开关失败。
HWRtcErrorCodeSetLocalAudioMuteFail	90000041	启停上行音频流失败	启停上行音频流失败。
HWRtcErrorCodeSetLocalVideoMuteFail	90000042	启停上行视频流失败	启停上行视频流失败。
HWRtcErrorCodeUserRemoved	90000043	用户被移除	用户被移除。
HWRtcErrorCodeRoomDismissed	90000044	房间被解散	房间被解散。
HWRtcErrorCodeSetupRemoteViewFail	90000045	设置远端View失败	设置远端View失败。
HWRtcErrorCodeLocalAudioDisableFail	90000056	当前未推音频流	当前未推音频流。
HWRtcErrorCodeRoleNotSupport	90000057	当前角色不支持该操作	当前角色不支持该操作。
HWRtcErrorCodeButt	90000058	置于最后	置于最后。
RTCErrorCode	90000060	发送CMD时，目标用户不存在	发送CMD时，目标用户不存在。
RTCErrorCode	90000061	CMD没有被启用	CMD没有被启用。

6.4.8 服务端错误码

当SDK运行出现网络、媒体相关等错误时，SDK无法自动恢复，需要APP干预或进行用户提示。该错误码由服务端产生，通过onError返回。

表 6-26 服务端错误码

错误码	描述	错误原因
RTC.10000001	内部错误	程序或环境问题
RTC.31000000	节点不存在	程序或环境问题
RTC.31000001	session校验失败	程序或环境问题
RTC.31000003	内部异常	程序或环境问题
RTC.31000004	认证失败	用户使用问题
RTC.31000005	请重试	用户使用问题
RTC.31000006	需要时钟同步	用户使用问题
RTC.31000007	请求资源不存在	程序或环境问题
RTC.32000000	服务异常	程序或环境问题
RTC.32000001	流号已满	程序或环境问题
RTC.32000002	SFU为空	程序或环境问题
RTC.32000004	下发流信息失败	程序或环境问题
RTC.32000005	添加适配器失败	程序或环境问题
RTC.32000006	添加路由失败	程序或环境问题
RTC.32000007	获取用户信息异常	程序或环境问题
RTC.32000010	选看用户不存在	程序或环境问题
RTC.32000011	音频速率参数非法	程序或环境问题
RTC.32000012	用户列表为空	程序或环境问题
RTC.32000013	非法请求参数	程序或环境问题
RTC.32000015	内部调用异常	程序或环境问题
RTC.32000016	内部调用异常	程序或环境问题
RTC.32000017	站点不存在	程序或环境问题
RTC.32000018	错误的加密算法	程序或环境问题
RTC.32000019	客户端媒体加密密钥base64解码失败	程序或环境问题
RTC.32000020	生成媒体加密密钥失败	程序或环境问题

错误码	描述	错误原因
RTC.32000021	下发加密信息异常	程序或环境问题
RTC.32000022	获取SFU的IP失败	程序或环境问题
RTC.32000024	内部调用异常	程序或环境问题
RTC.32000025	内部调用异常	程序或环境问题
RTC.32000028	不支持的操作	程序或环境问题
RTC.32000030	sfu资源不足	程序或环境问题
RTC.32000032	跨房数量超过上限	用户使用问题
RTC.32000033	不允许重复跨入同一房间	用户使用问题
RTC.32000034	稍后重试	程序或环境问题
RTC.33000000	服务异常	程序或环境问题
RTC.33000001	节点不存在	程序或环境问题
RTC.34000001	房间已满	用户使用问题
RTC.34000002	房间不存在	程序或环境问题
RTC.34000003	站点不存在	程序或环境问题
RTC.34000004	内部调用异常	程序或环境问题
RTC.34000006	用户不存在	用户使用问题
RTC.34000007	已存在辅流共享	用户使用问题

6.4.9 数据类型

本章节列出了iOS/macOS SDK的所有数据类型，您可以结合HWRtcEngine接口和回调进行开发。

表 6-27 数据类型

类型	描述
HWRtcUserInfo	用户信息
HWRtcJoinParam	入会信息
HWRtcVideoEncode	视频编码
HWRtcDeviceInfo	设备信息
HWRtcLogLevel	日志级别
HWRtcRole	用户角色

类型	描述
HWRtcMediaType	媒体类型
HWRtcVideoDisplayMode	视频渲染模式
HWRtcStreamType	流类型
HWRtcSpeakerModel	声音播放模式
HWRtcVideoRotation	视频角度
HWRtcAudioRoute	音频路由
HWRtcDeviceType	设备类型
HWRtcDeviceState	设备状态
HWRtcConnChangeReason	网络连接状态改变原因
HWRtcConnStateType	网络连接状态
HWRtcVideoCanvas	预览视图
HWRtcStatsInfo	卡顿统计信息
HWRtcVideoMirrorType	视频镜像类型
HWRtcRemoteAudioState	远端音频流状态
HWRtcRemoteAudioStateReason	远端音频流状态改变的原因
HWRtcRemoteVideoState	远端视频流状态
HWRtcRemoteVideoStateReason	远端视频流状态改变的原因
HWRtcLeaveReason	离开房间的原因
HWRtcMediaUsersVolumeInfo	发言人音量信息
HWRtcMediaDirection	媒体方向
HWRtcAudioFilePlayMode	音频播放类型
HWRtcAudioFrameType	音频格式
HWRtcVideoImageFormat	视频帧图片存储格式
HWRtcVideoImageBufferType	视频帧缓冲区存储类型
HwRtcImageBufferFormat	视频帧图片格式
HWRtcAudioFrame	音频数据信息
HWRtcVideoFrame	视频数据信息
HWRtcStartAudioFileParam	音频文件信息
HWRtcNetworkTestConfig	入会前网络检测配置

类型	描述
HWRtcNetworkTestResult	入会前网络检测回调结果
HWRtcNetworkTestState	入会前网络检测回调网络状态
HWRtcNetworkTestResultParam	入会前网络检测回调网络质量信息
HWRtcNetworkQualityLevel	入会前网络检测回调网络质量等级
HWRtcVideoStatsInfo	视频信息
HWRtcAudioStatsInfo	音频信息
HWRtcVideoStreamType	大小流类型
HWRtcVideoEncodeResolutionMode	视频编码比例模式
HWRtcGSensorMode	重力感应模式选择
HWRtcLocalAudioState	本地音频流状态
HWRtcLocalAudioStateReason	本地音频流状态发生变化原因
HWRtcLocalVideoState	本地视频流状态
HWRtcLocalVideoStateReason	本地视频流状态发生变化原因
HWRtcQualityInfo	网络质量信息
HWRtcRemoteAudioMode	远端音频模式
HRTCOnStats	会话统计信息
HWRtcAudioFileState	音频文件播放状态
HWRtcAudioFileReason	音频文件播放状态变化原因
HWRtcLocalVideoStats	本端视频统计回调
HWRtcRemoteVideoStats	远端视频统计回调
HWRtcLocalAudioStats	本端音频统计回调
HWRtcRemoteAudioStats	远端音频统计回调
HRTCMediaOptions	媒体选参
HWRtcBeautyOptions	美颜参数
HWRTCEncryptionConfig	端到端加密参数
HRTCUrlStatusList	rtmp推流回调url状态列表
HWRtcVideoOrientaion	方向（横竖屏）
HWRtcMediaConnChangeReason	媒体连接状态

类型	描述
HwRtcNetProxyConfig	代理参数的配置
HWRTCAudioOperateMode	采集数据回调的处理模式
RtmpConfigModel	RTMP推流设置参数模型
PublishStreamTypeModel	RTMP推流流类型模型
TranscodeConfigModel	RTMP推流模型
HWRtcStartAudioFileParam	音频文件参数
HWRtcAudioDeviceTestVolumeNotify	音量测试回调
HRTCRtmpUrlInfoModel	RTMP url数据模型
HWRTCCameraConfig	摄像头配置参数
HWRtcCameraDirection	摄像头方向
HWRtcVideoAuxiliarEncParam	辅流编码能力
HWRtcAudioDeviceTestVolumeNotify	音频设备测试回调数据
HwRtcNetProxyConfig	代理配置
HWRtcScreenShareSourceInfo	共享资源信息
HWRtcScreenShareParam	共享对象参数
HWRtcScreenShareType	共享类型枚举
HWRtcNetworkBandwidth	带宽设置参数
HWRtcRemoteMicState	麦克风设备状态
HRTCMultiRoomMediaRelayConfiguration	跨房配置
HRTCSrcMultiRoomMediaInfo	源房间信息
HRTCDstMultiRoomMediaInfo	目标房间信息
HRTCMultiRoomMediaRelayState	跨房状态
HRTCMultiRoomMediaRelayStateCode	跨房状态码

HWRtcUserInfo

表 6-28 用户信息

属性	类型	描述
userId	NSString	用户ID
userName	NSString	用户昵称
signature	NSString	鉴权签名，具体生成方法请参见17-接入鉴权
ctime	long long	鉴权时间戳
optionInfo	NSString	其他信息
role	**HWRtcRole**	用户角色

HWRtcJoinParam

表 6-29 入会参数

属性	类型	描述
userId	NSString	用户ID，支持最大长度64，支持数字、字母大小写、下划线、中线、"."字符
userName	NSString	用户昵称，支持最大长度128
ctime	long long	签名时间戳，单位秒，有signature时必选
authorization	NSString	签名，必填，鉴权私钥请在**应用管理**中获取，签名的具体生成方法请参见**接入鉴权**。支持最大长度为1024。
role	**HWRtcRole**	角色
optionalInfo	NSString	其他信息
roomId	NSString	房间Id，支持最大长度64，支持数字、字母大小写、下划线、中线字符
autoSubscribeVideo	bool	是否自动订阅视频
autoSubscribeAudio	bool	是否自动订阅音频
scenario	**HWRtcRemoteAudioMode**	使用的场景 • 0：主动订阅（默认） • 1：TopN（千人） • 2：P2P • 3：RTSA CMD自动订阅

HWRtcVideoEncode

表 6-30 编码设置

属性	类型	描述
streamFlag	HWRtcStreamType	视频编码类型，根据HWRtcStreamType和表57 不同分辨率下帧率和码率的推荐值设置需要的分辨率和宽高比
width	int	视频分辨率宽
height	int	视频分辨率高
frameRate	int	视频帧率，可以参考表6-82和表6-83进行设置
minFrameRate	int	视频最小帧率，大于0，小于frameRate
bitrate	int	视频码率，可以参考表6-82和表6-83进行设置
minBitrate	int	视频最小码率，大于0，小于bitrate
disableAdjustRes	bool	上行流是否自适应，推荐开启自适应（即disableAdjustRes赋NO）

HWRtcDeviceInfo

设备信息，macOS SDK使用。

表 6-31 设备信息

枚举值	类型	描述
deviceName	NSString	设备名称
deviceId	NSString	设备ID

HWRtcAudioQualityLevel

表 6-32 音频档位

枚举值	描述
HWRtcAudioQualityLevelDefault	默认值，表示使用采样率16KHZ、单声道、编码码率，最大值为30Kbps

HWRtcAudioSceneType

表 6-33 音频场景

枚举值	描述
HWRtcAudioSceneDefault	默认值，表示会议模式
HWRtcAudioSceneMusic	表示音乐模式

HWRtcLogLevel

表 6-34 日志级别

枚举值	描述
HWRtcLogLevelError	输出ERROR级别日志
HWRtcLogLevelWarning	输出WARNING级别日志
HWRtcLogLevelInfo	输出INFO级别日志
HWRtcLogLevelDebug	输出DEBUG级别日志

HWRtcEngineConfig

表 6-35 引擎配置

属性	类型	描述
appId	NSString	应用ID，只有App ID相同的应用程序才能进入同一个房间进行互动。appId获取方法请参见**应用管理**
countryCode	NSString	国家码，具体值请参见**国家码对照表**
enableHaTrace	BOOL	打点开关
muteAudioRoute	BOOL	是否禁音频路由
enableLog	BOOL	是否记录日志
logLevel	HWRtcLogLevel	日志级别，具体请参见**HWRtcLogLevel**，默认值为HWRtcLogLevelDebug，推荐使用HWRtcLogLevelDebug
logPath	NSString	日志路径，需调用方保证路径合法可用，rtc仅做基础校验

属性	类型	描述
logSize	int	日志大小，默认值10M，推荐10M

HWRtcRole

表 6-36 用户角色

枚举值	描述
HWRtcRoleJoiner	双向流角色，例如主播加入
HWRtcRolePlayer	接收流角色，例如观众

HWRtcMediaType

表 6-37 媒体类型

枚举值	描述
HWRtcMediaTypeAudio	只有音频流
HWRtcMediaTypeVideo	音频流+视频流

HWRtcVideoDisplayMode

表 6-38 渲染模式

枚举值	描述
HWRtcVideoDisplayModeFit	优先保证视频内容全部显示，视频尺寸等比缩放，直至视频窗口的一边与视窗边框对齐。如果视频长宽与显示窗口不同，视窗上未被填满的区域将被涂黑
HWRtcVideoDisplayModeHIDDEN,	优先保证视窗被填满，视频尺寸等比缩放，直至整个视窗被视频填满。如果视频长宽与显示窗口不同，多出的视频将被截掉
HWRtcVideoDisplayModeFill,	视频尺寸进行缩放和拉伸以充满显示视窗

HWRtcStreamType

表 6-39 流类型

枚举值	描述
HWRtcStreamTypeSD	标清
HWRtcStreamTypeHD	高清
HWRtcStreamTypeFHD	全高清

HWRtcSpeakerModel

表 6-40 播放声音模式

枚举值	描述
HWRtcSpeakerModelEarpiece	听筒模式
HWRtcSpeakerModelSpeaker	扬声器模式

HWRtcVideoRotation

表 6-41 视频角度

枚举值	描述
HWRtcVideoRotation0	视频角度0度
HWRtcVideoRotation90	视频角度90度
HWRtcVideoRotation180	视频角度180度
HWRtcVideoRotation270	视频角度270度

HWRtcAudioRoute

表 6-42 音频路由

枚举值	描述
HWRtcAudioRouteSpeaker	扬声器
HWRtcAudioRouteBluetooth	蓝牙
HWRtcAudioRouteReceiver	听筒
HWRtcAudioRouteHeadset	耳机

HWRtcDeviceType

设备类型，MAC SDK使用。

表 6-43 设备类型

枚举值	描述
HWRtcDeviceTypePlayback	音频播放设备
HWRtcDeviceTypeRecording	音频录制设备
HWRtcDeviceTypeVideoCapture	视频采集设备

HWRtcDeviceState

设备状态，MAC SDK使用。

表 6-44 设备状态

枚举值	描述
HWRtcDeviceStateActicve	设备状态活跃的
HWRtcDeviceStateDisabled	设备状态不使能的
HWRtcDeviceStateUnplugged	设备状态断开的

HWRtcConnChangeReason

表 6-45 网络连接状态改变原因

枚举值	描述
HWRtcConnChangeConnecting	正在连接
HWRtcConnChangeJoinSuccess	加入房间成功
HWRtcConnChangeReconnecting	重连中
HWRtcConnChangeReconnectSuccess	重连成功
HWRtcConnChangeJoinFailed	加入房间失败
HWRtcConnChangeReconncetFailed	重连失败
HWRtcConnChangeInterrupted	连接中断

枚举值	描述
HWRtcConnChangeKeepAliveTimeout	心跳超时
HWRtcConnChangeLeaveRoom	主动离开房间
HWRtcConnChangeJoinServerError	服务器异常
HWRtcConnChangeSFUBreakdown	sfu服务故障
HWRtcConnChangeJoinRoomAuthFailed	鉴权失败，appid或者签名错误
HWRtcConnChangeJoinRoomAuthRetry	鉴权重试
HWRtcConnChangeJoinRoomAuthClockSync	时钟同步
HWRtcConnChangeUrlNotRight	URL错误400
HWRtcConnChangeJoinRoomServiceUnreachable	服务不可达503
HWRtcConnChangeInternalError	内部错误
HWRtcConnChangeKickedOff	被踢
HWRtcConnChangeSignatureExpired	签名过期

HWRtcConnStateType

表 6-46 网络连接状态

枚举值	描述
HWRtcConnStateTypeDisconnected	连接断开
HWRtcConnStateTypeConnecting	建立网络连接中
HWRtcConnStateTypeConnected	网络连接成功
HWRtcConnStateTypeReconnecting	重新建立网络连接中

枚举值	描述
HWRtcConnStateTypeFailed	网络连接失败
HWRtcConnStateTypeLost	失去网络10秒
HWRtcConnStateTypeInterrupted	SDK在和服务器建立连接后，失去网络连接超过4秒

HWRtcVideoCanvas

表 6-47 预览视图

属性	描述
@property (strong, nonatomic) UIView* _Nullable view;	视频显示视图，使用跟系统的UIView/NSView一样
@property (assign, nonatomic) NSString uid;	视图的用户标识

HWRtcStatsInfo

表 6-48 卡顿统计信息

属性	描述
@property (strong, nonatomic) NSString *roomId;	房间ID
@property (strong, nonatomic) NSString *userId;	用户ID
@property (assign, nonatomic) long long mildlyFrozenCounts;	400ms卡顿次数
@property (assign, nonatomic) long long severelyFrozenCounts;	超过1s卡顿次数
@property (assign, nonatomic) long long totalMildlyFrozenTime;	400ms卡顿总时长
@property (assign, nonatomic) long long totalSeverelyFrozenTime;	1s卡顿总时长

属性	描述
@property (assign, nonatomic) long long totalActiveTime;	总时间，包括每一路选看时间总和

HWRtcVideoMirrorType

表 6-49 视频镜像类型

枚举值	描述
HWRtcVideoMirrorTypeAuto	SDK决定镜像方式：前置摄像头镜像，后置摄像头不镜像
HWRtcVideoMirrorTypeEnable	前置摄像头和后置摄像头都镜像
HWRtcVideoMirrorTypeDisable	前置摄像头和后置摄像头都不镜像

HWRtcRemoteAudioState

表 6-50 远端音频状态

枚举值	描述
HWRtcRemoteAudioStateStopped	远端音频流关闭发送
HWRtcRemoteAudioStateDecoding	远端音频流正常编码发送
HWRtcRemoteAudioStateFirstDecoded	远端音频流首包解码

HWRtcRemoteAudioStateReason

表 6-51 远端音频状态变化原因

枚举	描述
HWRtcRemoteAudioStateReasonRemoteOffline	远端用户离线
HWRtcRemoteAudioStateReasonRemoteMuted	远端用户停止音频流发送
HWRtcRemoteAudioStateReasonRemoteUnmuted	远端用户开启音频流发送

枚举	描述
HWRtcRemoteAudioStateReasonRemoteFirstDecoded	远端用户音频首包解码

HWRtcRemoteVideoState

表 6-52 远端视频状态

枚举值	描述
HWRtcRemoteVideoStateStopped	远端视频流关闭发送
HWRtcRemoteVideoStateDecoding	远端视频流正常编码发送

HWRtcRemoteVideoStateReason

表 6-53 远端视频状态变化原因

枚举值	描述
HWRtcRemoteVideoStateReasonRemoteOffline	远端用户离线
HWRtcRemoteVideoStateReasonRemoteMuted	远端用户停止视频流发送
HWRtcRemoteVideoStateReasonRemoteUnmuted	远端用户开启视频流发送

HWRtcLeaveReason

表 6-54 离开房间原因

枚举值	描述
HWRtcLeaveReasonUserLeaveRoom	用户主动离开
HWRtcLeaveReasonServerError	服务器异常
HWRtcLeaveReasonBreakdown	sfu服务故障

枚举值	描述
HWRtcLeaveReasonServiceUnreachable	服务不可达
HWRtcLeaveReasonInternalError	内部错误
HWRtcLeaveReasonKickedOff	被踢
HWRtcLeaveReasonAuthorizationExpired	签名过期
HWRtcLeaveReasonReconnectFailed	重连超时

HWRtcMediaUsersVolumeInfo

表 6-55 发言人音量信息

属性	描述
@property (strong, nonatomic) NSString *userId;	用户ID
@property (assign, nonatomic) int volume;	音量

HWRtcMediaDirection

表 6-56 媒体方向

枚举值	描述
HWRtcMediaDirectionLocal	本地
HWRtcMediaDirectionRemote	远端

HWRtcAudioFilePlayMode

表 6-57 播放类型

枚举	描述
HWRtcAudioFileLocalPlayMode	本地播放模式

枚举	描述
HWRtcAudioFileRemotePlayMode	远端播放模式

HWRtcAudioFrameType

表 6-58 音频帧格式

枚举	描述
HWRtcAudioFrameTypePCM16	量化为16比特

HWRtcVideoImageFormat

表 6-59 视频帧图片存储格式

枚举	描述
HWRtcVideoImageFormatYUV420P	采样比例为4：2：0，存储方式为I420
HWRtcVideoImageFormatYUV422I	采样比例4：2：0
HWRtcVideoImageFormatRGB24	格式为RGB
HWRtcVideoImageFormatRGBA	格式为RGBA

HWRtcVideoImageBufferType

表 6-60 视频帧缓冲区存储类型

枚举	描述
HWRtcVideoImageBufferByteArray	Byte Array类型，对应HWRtcVideoImageFormat中的YUV、RGBA格式

HwRtcImageBufferFormat

表 6-61 视频帧图片格式

属性	描述
@property HWRtcVideoImageFormat format;	视频帧图片存储格式，具体请参见 **HWRtcVideoImageFormat**
@property HWRtcVideoImageBufferType bufferType;	视频帧缓冲区存储类型，具体请参见 **HWRtcVideoImageBufferType**

HWRtcAudioFrame

表 6-62 音频信息

属性	描述
@property (nonatomic, assign) HWRtcAudioFrameType frameType;	音频帧类型，具体请参见 **HWRtcAudioFrameType**
@property (nonatomic, assign) NSUInteger samples;	每个声道的采样点数，iSamples=iSamplesPerSec * 10ms /1000
@property (nonatomic, assign) NSUInteger samplesPerSec;	采样率
@property (nonatomic, assign) NSUInteger bytesPerSample; // bitsPerSample	每个采样点的字节数，pcm数据一般2字节
@property (nonatomic, assign) NSUInteger channels;	声道数
@property (strong, nonatomic) NSData * _Nullable data;	数据缓冲区，长度 =samples*bytesPerSample*channels

HWRtcVideoFrame

表 6-63 视频帧信息

属性	描述
@property (assign, nonatomic) HWRtcVideoImageFormat format;	图像格式 ，具体请参见 **HWRtcVideoImageFormat**
@property (assign, nonatomic) NSUInteger width;	图像宽度，作为输入时，范围为[90,1920]，必须是4的整数倍
@property (assign, nonatomic) NSUInteger height;	图像高度，作为输入时，范围为[90,1200]，必须是2的整数倍
@property (strong, nonatomic) NSData * _Nullable data;	存储视频数据的buf地址
@property (assign, nonatomic) NSUInteger dataLen;	视频数据的长度，单位为Byte
@property (strong, nonatomic) NSString *userId;	被选看的远端用户本地推流不需要赋值

HWRtcStartAudioFileParam

表 6-64 音频文件信息

属性	描述
@property (nonatomic, strong) NSString * _Nonnull fullFilePath;	文件路径
@property (nonatomic, assign) HWRtcAudioFilePlayMode playMode;	播放类型，具体请参见 **HWRtcAudioFilePlayMode**
@property (nonatomic, assign) NSUInteger cycle;	循环次数
@property (nonatomic, assign) NSUInteger replace;	远端模式下面是否需要和麦克风做混音

HWRtcNetworkTestConfig

表 6-65 入会前网络检测配置

属性	描述
@property (strong, nonatomic) NSString * _Nonnull userId	用户id
@property (strong, nonatomic) NSString * _Nonnull roomId	房间ID，roomid建议由userid+随机数组成
@property (strong, nonatomic) NSString *signature	必选，签名信息
@property (nonatomic, assign) long long ctime	必选，系统时间
@property (assign, nonatomic) BOOL enableUplinkTest	启动上行网络检测开关
@property (assign, nonatomic) BOOL enableDownlinkTest	启动下行网络检测开关
@property (nonatomic, assign) NSUInteger expectedUplinkBitrate	用户期望的最高发送码率，单位为bps，范围为0以及[100000, 5000000]，设为0表示由SDK指定最高码率
@property (nonatomic, assign) NSUInteger expectedDownlinkBitrate	用户期望的最高接收码率，单位为bps，范围为0以及[100000, 5000000]，设为0表示由SDK指定最高码率

HWRtcNetworkTestResult

表 6-66

属性	描述
@property (assign, nonatomic) HWRtcNetworkTestState state	检测网络状态，具体请参见 **HWRtcNetworkTestState**
@property (nonatomic, strong) HWRtcNetworkTestResultParam *uplinkResult	上行网络检测状态，具体请参见 **HWRtcNetworkTestResultParam**

属性	描述
@property (nonatomic, strong) HWRtcNetworkTestResultParam * downlinkResult	下行网络检测状态，具体请参见 HWRtcNetworkTestResultParam

HWRtcNetworkTestState

表 6-67 检测网络测试状态

属性	描述
HWRtcNetworkTestOK	成功
HWRtcNetworkTestFAIL	失败

HWRtcNetworkTestResultParam

表 6-68 检测网络回调信息

属性	描述
@property (nonatomic, assign) NSInteger bitRate	码率
@property (nonatomic, assign) NSInteger packetLoss	丢包
@property (nonatomic, assign) NSInteger delay	延时
@property (nonatomic, assign) NSInteger jitter	抖动

HWRtcVideoStatsInfo

表 6-69 视频信息

属性	描述
@(strong, nonatomic) NSString *userId;	用户ID（本地视频信息为空）
@ (assign, nonatomic) NSUInteger width;	视频宽

属性	描述
@ (assign, nonatomic) NSUInteger height;	视频高
@ (assign, nonatomic) NSUInteger bitRate;	视频码率
@ (assign, nonatomic) NSUInteger frameRate;	视频帧率
@ (assign, nonatomic) NSUInteger packetLoss;	视频丢包率
@ (assign, nonatomic) NSUInteger delay;	视频时延
@ (assign, nonatomic) NSUInteger jitter;	视频抖动

HWRtcAudioStatsInfo

表 6-70 音频信息

属性	描述
@(strong, nonatomic) NSString *userId;	用户ID（本地音频信息为空）
@ (assign, nonatomic) NSUInteger sampleRate;	音频采样率
@ (assign, nonatomic) NSUInteger channels;	音频声道数
@ (assign, nonatomic) NSUInteger bitRate;	音频码率
@ (assign, nonatomic) NSUInteger packetLoss;	音频丢包率
@ (assign, nonatomic) NSUInteger delay;	音频时延
@ (assign, nonatomic) NSUInteger jitter;	音频抖动

HWRtcNetworkQualityLevel

表 6-71 检测网络质量等级

属性	描述
HWRtcNetworkQualityUnknown	网络质量未知
HWRtcNetworkQualityExcellent	网络质量非常好
HWRtcNetworkQualityGood	网络质量好
HWRtcNetworkQualityPoor	网络质量一般
HWRtcNetworkQualityBad	网络质量差
HWRtcNetworkQualityVbad	网络质量非常差

HWRtcVideoStreamType

表 6-72 大小流类型

属性	描述
HWRtcVideoStreamTypeBig	大流类型
HWRtcVideoStreamTypeSmall	小流类型

HWRtcVideoEncodeResolutionMode

表 6-73 视频编码分辨率比例模式

属性	描述
HWRtcVideoEncodeResolutionModeNone	不固定比例
HWRtcVideoEncodeResolutionModeConstRatio	固定比例

HWRtcGSensorMode

表 6-74 重力感应模式

属性	描述
HWRtcGSensorModeDisable	关闭重力感应

属性	描述
HWRtcGSensorModeUIAutoLayout	开启重力感应 **注意**：SDK不会根据陀螺仪自动调整本地View的画面方向，而是需要您的APP开启了重力感应进行适配

HWRtcLocalAudioState

表 6-75 本地音频流状态

属性	描述
HWRtcLocalAudioStateStopped	本地音频流默认初始状态
HWRtcLocalAudioStateRecording	本地音频流录制设备启动成功
HWRtcLocalAudioStateFailed	本地音频流启动失败

HWRtcLocalAudioStateReason

表 6-76 本地音频流状态变化原因

属性	描述
HWRtcLocalAudioReasonErrorOk	本地音频流状态正常
HWRtcLocalAudioReasonErrorFailure	本地音频流出错原因不明确
HWRtcLocalAudioReasonErrorRecordFailure	本地音频流录制失败，建议您检查录制设备是否正常工作
HWRtcLocalAudioReasonErrorStopFailure	关闭采集失败
HWRtcLocalAudioReasonErrorAccessDenied	音频设备API无法访问，可能是设备隐私权限设置问题
HWRtcLocalAudioReasonErrorOnExclusiveMode	音频设备处于独占模式，且被其他应用独占
HWRtcLocalAudioReasonErrorMmsyserrInvalparam	音频设备API非法参数（仅适用于windows）
HWRtcLocalAudioReasonErrorMmsyserrNodriver	音频设备API返回无驱动，需要用户升级驱动（仅适用于windows）
HWRtcLocalAudioReasonErrorAudioServerNotRunning	用户windows audio服务未启动，或者启动失败（仅适用于windows）

属性	描述
HWRtcLocalAudioReasonErrorNoDevice	没有设备（仅适用于windows）
HWRtcLocalAudioReasonErrorRestartFailed	扬声器播放无数据，重启失败

HWRtcLocalVideoState

表 6-77 本地视频流状态

属性	描述
HWRtcLocalVideoStateStopped	本地视频流默认初始状态
HWRtcLocalVideoStateCapturing	本地视频流采集设备启动成功
HWRtcLocalVideoStateFailed	本地视频流启动失败

HWRtcLocalVideoStateReason

表 6-78 本地视频流状态变化原因

属性	描述
HWRtcLocalVideoReasonErrorOk	本地视频流状态正常
HWRtcLocalVideoReasonErrorFailure	本地视频流出错原因不明确
HWRtcLocalVideoReasonErrorCaptureFailure	本地视频流录制失败，建议您检查录制设备是否正常工作
HWRtcLocalVideoReasonErrorStopFailure	关闭采集失败
HWRtcLocalVideoReasonErrorCaptureDeviceNoPermission	没有摄像头权限
HWRtcLocalVideoReasonErrorCaptureDeviceBusy	摄像头设备已占用
HWRtcLocalVideoReasonErrorCaptureAppInBackground	应用处于后台
HWRtcLocalVideoReasonErrorCaptureOpenCameraFailed	打开摄像头设备失败（仅支持iOS）
HWRtcLocalVideoReasonErrorCaptureMultipleForegroundApp	应用窗口处于侧拉、分屏、画中画模式（仅支持iOS）

属性	描述
HWRtcLocalVideoReasonErrorCaptureMultipleDeviceDisconnected	本地视频采集设备未连接（仅支持macOS）

HWRtcQualityInfo

表 6-79 网络质量信息

属性	描述
@property (strong, nonatomic) NSString *userId;	用户ID
@property (assign, nonatomic) HWRtcNetworkQualityLevel level;	网络质量等级
@property (assign, nonatomic) NSUInteger width;	宽度
@property (assign, nonatomic) NSUInteger height;	高度

HWRtcNetworkQualityLevel

表 6-80 网络质量等级

属性	描述
HWRtcNetworkQualityUnknown	网络质量未知
HWRtcNetworkQualityExcellent	网络质量非常好
HWRtcNetworkQualityGood	网络质量好
HWRtcNetworkQualityPoor	网络质量一般
HWRtcNetworkQualityBad	网络质量差
HWRtcNetworkQualityVbad	网络质量非常差

HWRtcRemoteAudioMode

表 6-81 远端音频流收流模式

枚举值	描述
HWRtcRemoteAudioSubscribe= 0	订阅模式（自主订阅）

枚举值	描述
HWRtcRemoteAudioTopThre= 1	TopN模式（收音量最大的三路流）
HRTC_REMOTE_AUDIO_P2P= 2	P2P模式
HRTC_REMOTE_AUDIO_RTSA_CMD =3	RTSA-CMD模式

表 6-82 不同分辨率下帧率和码率的推荐值

分辨率	分辨率类型	比例	最小帧率（fps）	最大帧率（fps）	最小码率	最大码率
320 X 180	SD	16:9	10	30	80	600
480 X 270	HD	16:9	10	30	160	1050
640 X 360	HD	16:9	10	30	200	1700
800 X 450	FHD	16:9	10	30	300	2100
960 X 540	FHD	16:9	10	30	400	2400
1120 X 630	FHD	16:9	10	30	450	2800
1280 X 720	FHD	16:9	10	30	500	4000
120 X 90	LD	4:3	10	30	64	240
160 X 120	SD	4:3	10	30	64	270
240 X 180	SD	4:3	10	30	80	450
320 X 240	HD	4:3	10	30	100	600
400 X 300	HD	4:3	10	30	200	900
480 X 360	HD	4:3	10	30	200	1000
640 X 480	FHD	4:3	10	30	250	1800
960 X 720	FHD	4:3	10	30	450	3000

表 6-83 不同场景下帧率和码率的推荐值

分辨率	推荐帧率	通信场景推荐码率	直播场景推荐码率
160 X 90	15	90	180
320 X 180	15	200	400
480 X 270	15	350	700
640 X 360	15	450	900

分辨率	推荐帧率	通信场景推荐码率	直播场景推荐码率
640 X 360	30	850	1700
800 X 450	15	700	1400
800 X 450	30	1050	2100
960 X 540	15	850	1700
960 X 540	30	1200	2400
1120 X 630	15	950	1900
1120 X 630	30	1400	2800
1280 X 720	15	1200	2400
1280 X 720	30	2000	4000
120 X 90	15	80	160
160 X 120	15	90	180
240 X 180	15	150	300
320 X 240	15	200	400
400 X 300	15	300	600
480 X 360	15	350	700
480 X 360	30	500	1000
640 X 480	15	600	1200
640 X 480	30	900	1800
960 X 720	15	1000	2000
960 X 720	30	1500	3000

HRTCOnStats

表 6-84 引擎创建相关参数

属性	类型	描述
cpuAppUsage	double	app的cpu利用率，单位(%)
cpuTotalUsage	double	cpu总利用率，单位(%)
memoryAppUsageInKbytes	int	app占用内存，单位KB

属性	类型	描述
memoryAppUsageRatio	double	app占用的内存率，单位(%)
memoryTotalUsageRatio	double	总的内存利用率，单位(%）
gatewayRtt	int	到本地网关的延迟，单位ms
sendBytes	long	总的发送字节数，单位bytes
sendVideoBytes	long	视频的发送字节数，单位bytes
sendAudioBytes	long	音频的发送字节数，单位bytes
receiveBytes	long	总的接收字节数，单位bytes
receiveVideoBytes	long	视频的接收字节数，单位bytes
receiveAudioBytes	long	音频的接收字节数，单位bytes
sendBitRate	int	总的发送比特率，单位Kbps
sendVideoBitRate	int	视频的发送比特率，单位Kbps
sendAudioBitRate	int	音频的发送比特率，单位Kbps
receiveBitRate	int	总的接收比特率，单位Kbps
receiveVideoBitRate	int	视频的接收比特率，单位Kbps
receiveAudioBitRate	int	音频的接收比特率，单位Kbps
sendLossRate	int	发送丢包率，单位(%)
receiveLossRate	int	接收丢包率，单位(%)
lastmileDelay	int	到服务器的延迟，单位ms

HWRtcAudioFileState

表 6-85 音频文件播放状态

属性	描述
HWRtcAudioFileOpenCompleted	成功打开音频文件

属性	描述
HWRtcAudioFileOpening	正在打开音频文件
HWRtcAudioFileIdle	音频文件播放就绪
HWRtcAudioFilePlaying	音频文件播放中
HWRtcAudioFilePlayCompleted	音频文件播放完成
HWRtcAudioFilePaused	音频文件暂停播放
HWRtcAudioFileStopped	音频文件停止播放
HWRtcAudioFileFailed	音频文件播放失败
HWRtcAudioFilePositionUpdate	音频文件播放进度更新
HWRtcAudioFileStateUnknown	音频文件播放状态未知

HWRtcAudioFileReason

表 6-86 音频播放状态变化原因

枚举值	描述
HWRtcAudioFileReasonNone	没有错误
HWRtcAudioFileReasonUrlNotFound	未找到URL
HWRtcAudioFileReasonCodecNotSupported	解码器不支持该编码
HWRtcAudioFileReasonInvalidArguments	非法参数
HWRtcAudioFileReasonSrcBufferUnderflow	播放缓冲区数据不足
HWRtcAudioFileReasonInterval	内部错误
HWRtcAudioFileReasonInvalidState	播放器内部状态错误
HWRtcAudioFileReasonNoResource	没有该资源
HWRtcAudioFileReasonObjNotInitialized	对象未初始化
HWRtcAudioFileReasonInvalidConnectionState	播放器与服务器连接无效
HWRtcAudioFileReasonUnknowdStreamType	未知的媒体流类型
HWRtcAudioFileReasonVideoRenderFailed	渲染失败

枚举值	描述
HWRtcAudioFileReasonInvalidMediaSource	无效的媒体资源
HWRtcAudioFileReasonUnknown	状态未知

HWRtcLocalVideoStats

表 6-87 本端视频统计回调

枚举值	描述
@(strong, nonatomic) NSString *userId;	用户ID（本地视频信息为空）
@ (assign, nonatomic) NSUInteger width;	视频宽
@ (assign, nonatomic) NSUInteger height;	视频高
@ (assign, nonatomic) NSUInteger bitRate;	视频码率
@ (assign, nonatomic) NSUInteger frameRate;	视频帧率
@ (assign, nonatomic) NSUInteger packetLoss;	视频丢包率
@ (assign, nonatomic) NSUInteger delay;	视频时延
@ (assign, nonatomic) NSUInteger jitter;	视频抖动
@ (assign, nonatomic) long long bytes;	字节数
@ (assign, nonatomic) NSUInteger sendFrameRate;	实际发送帧率，单位：fps

HWRtcRemoteVideoStats

表 6-88 远端视频统计回调

枚举值	描述
@(strong, nonatomic) NSString *userId;	用户ID（本地视频信息为空）
@ (assign, nonatomic) NSUInteger width;	视频宽

枚举值	描述
@ (assign, nonatomic) NSUInteger height;	视频高
@ (assign, nonatomic) NSUInteger bitRate;	视频码率
@ (assign, nonatomic) NSUInteger frameRate;	视频帧率
@ (assign, nonatomic) NSUInteger packetLoss;	视频丢包率
@ (assign, nonatomic) NSUInteger delay;	视频时延
@ (assign, nonatomic) NSUInteger jitter;	视频抖动
@ (assign, nonatomic) long long bytes;	字节数
@ (assign, nonatomic) NSUInteger sendFrameRate;	实际发送帧率，单位：fps
@ (assign, nonatomic) NSUInteger rendererOutputFrameRate;	渲染帧率，单位：fps
@ (assign, nonatomic) NSUInteger totalFrozenTime;	远端用户在加入房间后到离开房间前，发生视频卡顿的累计时长，单位：ms
@ (assign, nonatomic) NSUInteger frozenRate;	远端用户在加入房间后到离开房间前，发生视频卡顿的累计时长占视频总有效时长的百分比，单位：%

HWRtcLocalAudioStats

表 6-89 本端音频统计回调

属性	描述
@(strong, nonatomic) NSString *userId;	用户ID（本地音频信息为空）
@ (assign, nonatomic) NSUInteger sampleRate;	音频采样率
@ (assign, nonatomic) NSUInteger channels;	音频声道数
@ (assign, nonatomic) NSUInteger bitRate;	音频码率
@ (assign, nonatomic) NSUInteger packetLoss;	音频丢包率

属性	描述
@ (assign, nonatomic) NSUInteger delay;	音频时延
@ (assign, nonatomic) NSUInteger jitter;	音频抖动

HWRtcRemoteAudioStats

表 6-90 远端音频统计回调

属性	描述
@(strong, nonatomic) NSString *userId;	用户ID（本地音频信息为空）
@ (assign, nonatomic) NSUInteger sampleRate;	音频采样率
@ (assign, nonatomic) NSUInteger channels;	音频声道数
@ (assign, nonatomic) NSUInteger bitRate;	音频码率
@ (assign, nonatomic) NSUInteger packetLoss;	音频丢包率
@ (assign, nonatomic) NSUInteger delay;	音频时延
@ (assign, nonatomic) NSUInteger jitter;	音频抖动
@ (assign, nonatomic) NSUInteger totalFrozenTime;	远端用户在加入房间后到离开房间前，发生音频卡顿的累计时长，单位：ms **说明**：仅onRemoteAudioStatsNotify回调中携带此数据
@ (assign, nonatomic) NSUInteger frozenRate;	远端用户在加入房间后到离开房间前，发生音频卡顿的累计时长占音频总有效时长的百分比，单位：% **说明**：仅onRemoteAudioStatsNotify回调中携带此数据

HRTCSfuType

表 6-91 Sfu 类型

枚举值	描述
HRTC_SFU_TYPE_PUBLIC_NETWORK	公网sfu资源

HRTCMediaOptions

表 6-92 媒体选参

属性	类型	描述
autoSubscribeAudio	boolean	自动选看音频
autoSubscribeVideo	boolean	自动选看视频

HWRtcBeautyOptions

表 6-93 美颜选参

属性	类型	描述
buffingLevel	float	磨皮档位，取0-1浮点数。默认值0.5，推荐使用默认值0.5
complexionAlpha	float	肤色档位，取0-1浮点数。默认值0.5，推荐使用默认值0.5

HWRTCEncryptionConfig

表 6-94 加密配置

函数&属性	描述
@(strong, nonatomic) HWRTCCryptionMode cryptionMode;	HWRTCCryptionMode加密模式
@(assign, nonatomic) HWRTCSuiteType suitType;	HWRTCSuiteType加密算法，仅模式HWRTCCryptionModeAuthenticationSdk需要

函数&属性	描述
@ (assign, nonatomic) NSString *cryptionSec;	加密密钥，仅模式HWRTCCryptionModeAuthenticationSdk需要设置
@ (assign, nonatomic) HWRTCCryptionSecFormat cryptionSecFormat	HWRTCCryptionSecFormat ，密钥格式，当前只支持16进制字符串

HWRTCCryptionMode

表 6-95 加密模式

枚举值	描述
HWRTCCryptionModeDefault	不开启端到端加密，此时srtp认证（包校验）+加密
HWRTCCryptionModeAuthenticationSdk	开启端到端加密，srtp只认证（包校验），sdk内部加密, 必现配置key
HWRTCCryptionModeAuthenticationApp	开启端到端加密，srtp只认证（包校验），应用层加密，需注册回调

HWRTCSuiteType

表 6-96 加密密钥

函数&属性	描述
HWRTCSuiteType128Ctr	加密密钥，仅模式HWRTCCryptionModeAuthenticationSdk需要设置

HWRTCCryptionSecFormat

表 6-97 密钥格式

枚举值	描述
HWRTC_HEX_STRING	16进制字符串格式。当前只支持此格式。

HRTCUrlStatusList

表 6-98 HRTCRtmpUrlInfo

属性	描述
char url[1025]	url字符串
int status	状态码
int errCode	错误码

HWRtcMediaConnChangeReason

表 6-99 媒体连接状态

属性	描述
HWRtcMediaConnChangedConnected	连接成功
HWRtcMediaConnChangedNATFailed	NAT未打通

HWRtcVideoOrientaion

表 6-100 方向（横竖屏）

枚举值	描述
HRTC_VIDEO_ORIENTATION__LANSCAPE	横屏
HRTC_VIDEO_ORIENTATION__PORTRAIT	竖屏

HwRtcNetProxyConfig

表 6-101 代理参数的配置

属性	类型	描述
autoNetProxy	BOOL	是否开启自动代理
address	NSString	代理的地址
port	NSInteger	代理端口
name	NSString	代理认证的账号名
pwd	NSString	代理认证的账号密码

HWRTCAudioOperateMode

表 6-102 采集数据回调的处理模式

属性	描述
enum HWRTCAudioOperateMode.HWRTC_AUDIO_OPERATE_READ_AND_WRITE	可读可写模式

RtmpConfigModel

表 6-103 RTMP 推流设置参数模型

属性	类型	描述
width	int	旁路推流的输出视频流的总宽度，单位为px。360为默认值，取值范围为[64-1920]
height	int	旁路推流的输出视频流的总高度，单位为px。640为默认值，取值范围为[64-1920]
videoBitrate	int	旁路推流的输出视频的码率，单位为Kbps。默认值为400Kbps，取值范围[32-2760]，
videoFramerate	int	旁路推流的输出视频的帧率，单位为fps。默认值为15，取值范围为 [10,30]
videoGop	int	用于旁路直播的输出视频的GOP，单位为帧。默认值为30帧，取值范围为[1-300]
audioSampleRate	int	用于旁路直播的输出音频的采样率，默认16000，取值范围为[16000-96000]
audioBitrate	int	旁路直播的输出音频的码率。单位为Kbps，默认值为48，最大值为128，取值范围为[1-128]
audioChannels	int	旁路直播的输出音频的声道数，默认1，取值范围为[1-5]
tmplate	int	0表示悬浮，1表示九宫格，2表示屏幕分享，默认为0

PublishStreamTypeModel

表 6-104 RTMP 推流流类型模型

属性	类型	描述
main	bool	是否推大流
slides	bool	是否推小流
desktop	bool	是否推桌面流
audio	bool	是否推音频流

TranscodeConfigModel

表 6-105 RTMP 推流模型

属性	类型	描述
config	RtmpConfigModel	混流的配置信息
mapPublishStreamTypes	NSDictionary<NSString *, PublishStreamTypeModel *>	混流用户的流信息
userInfos	NSArray<NSString *>	混流用户信息

HWRtcStartAudioFileParam

表 6-106 音频文件参数

属性	类型	描述
fullFilePath	NSString	音频文件路径
playMode	**HWRtcAudioFilePlayMode**	播放模式
cycle	NSUInteger	音频播放次数，0表示无限循环
replace	NSUInteger	是否用音频文件替换麦克风采集的声音。1表示只使用音频文件发送到远端，0表示将本地麦克风采集并和音频文件混音后发送到远端

HWRtcAudioDeviceTestVolumeNotify

表 6-107 音量测试回调

属性	类型	描述
recordVolume	NSInteger	麦克风音量值0-100
playbackVolume	NSInteger	扬声器音量值0-100

HRTCRtmpUrlInfoModel

表 6-108 RTMP url 数据模型

属性	类型	描述
url	NSString	url字符串
status	int	状态码
errCode	int	错误码

HWRTCCameraConfig

表 6-109 摄像头配置参数

属性	类型	描述
direction	HWRtcCameraDirection	摄像头方向枚举，具体请参见 HWRtcCameraDirection

HWRtcCameraDirection

表 6-110 摄像头方向枚举

枚举	描述
HWRtcCameraDirectionRear	后置摄像头
HWRtcCameraDirectionFront	前置摄像头

HWRtcVideoAuxiliarEncParam

表 6-111 辅流编码能力

属性	类型	描述
frameRate	NSInteger	编码帧率，推荐帧率15帧
width	NSInteger	编码辅流宽。（iOS下根据宽，根据高按比例缩放）
height	NSInteger	编码辅流高。（iOS下如果屏幕宽度不超过1080，则视频源宽为720，否则为1080）
bitrate	NSInteger	码率，具体设置参考**表6-82**和**表6-83**。

HWRtcAudioDeviceTestVolumeNotify

表 6-112 HWRtcAudioDeviceTestVolumeNotif

属性	类型	描述
recordVolume	NSInteger	麦克风音量
playbackVolume	NSInteger	扬声器音量

HwRtcNetProxyConfig

表 6-113 代理参数的配置

属性	类型	描述
autoNetProxy	BOOL	是否开启自动代理
address	NSString	代理的地址
port	NSInteger	代理端口
name	NSString	代理认证的账号名
pwd	NSString	代理认证的账号密码

HWRtcScreenShareSourceInfo

表 6-114 共享资源信息

属性	类型	描述
sourceId	NSInteger	采集源ID

属性	类型	描述
sourceName	NSString	采集源名称
type	HWRtcScreenShareType	共享类型，具体请参见 **HWRtcScreenShareType**
icon	void*	预留字段，暂不支持
rect	NSRect	采集源区域

HWRtcScreenShareParam

表 6-115 共享对象参数

属性	类型	描述
type	HWRtcScreenShareType	共享类型，具体请参见 **HWRtcScreenShareType**
viewID	NSInteger	采集源ID
bCaptureMouse	BOOL	MAC不支持
rect	NSRect	采集源区域
sourceName	NSString	采集源名称

HWRtcScreenShareType

表 6-116 共享类型

枚举值	描述
HWRtcScreenShareTypeDesktop	桌面共享
HWRtcScreenShareTypeWindow	窗口共享

HWRtcVideoRemoteView

表 6-117 单个远端数据类型

属性	类型	描述
videoCanvas	**HWRtcVideoCanvas**	视频画布对象属性
streamType	**HWRtcStreamType**	要订阅的编码类型

属性	类型	描述
disableAdjustRes	int	下行码率自适应，默认关闭
minResolution	HWRtcStreamType	最小编码类型

HWRtcNetworkBandwidth

表 6-118 上下行带宽

属性	类型	描述
maxBandwidth	NSInteger	带宽上限。单位：KB。有效范围为[3072, 51200]，即3M~50M。

HRTCMultiRoomMediaRelayConfiguration

表 6-119 HRTCMultiRoomMediaRelayConfiguration

属性	类型	描述
srcRoomMediaInfo	**HRTCSrcMultiRoomMediaInfo**	源房间的鉴权信息
destRoomMediaInfo	**HRTCDstMultiRoomMediaInfo**	目的跨房的房间信息以及鉴权信息

HRTCSrcMultiRoomMediaInfo

表 6-120 HRTCSrcMultiRoomMediaInfo

属性	类型	描述
authorization	const char*	源房间的鉴权信息
userId	const char*	源房间的用户名
roomId	const char*	源房间的房间号
ctime	long long	鉴权时间信息

HRTCDstMultiRoomMediaInfo

表 6-121 HRTCDstMultiRoomMediaInfo

属性	类型	描述
authorization	const char*	目标跨房的鉴权信息
userId	const char*	目标跨房的虚拟用户名
roomId	const char*	目标跨房房间号
userRole	HRTCRoleType	跨房角色
ctime	long long	鉴权时间信息

HRTCMultiRoomMediaRelayState

表 6-122 HRTCMultiRoomMediaRelayState

属性	描述
HRTC_MULTI_ROOM_MEDIA_RELAY_STATE_IDLE	就绪状态
HRTC_MULTI_ROOM_MEDIA_RELAY_STATE_CONNECTING	正在连接
HRTC_MULTI_ROOM_MEDIA_RELAY_STATE_RUNNING	主播成功加入目标房间
HRTC_MULTI_ROOM_MEDIA_RELAY_STATE_FAILURE	发生异常

HRTCMultiRoomMediaRelayStateCode

表 6-123 HRTCMultiRoomMediaRelayStateCode

属性	描述
HRTC_MULTI_ROOM_MEDIA_RELAY_OK	正常状态
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_SERVER_NO_RESPONSE	服务端无响应
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_INTERNAL_ERROR	服务器内部出错
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_USER_OVER_LIMIT	用户跨房超出限制数

属性	描述
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_OVER_LIMIT	房间跨房用户超出限制数
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_REQ_EMPTY	跨房请求消息体为空
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_OPERATION_CONFLICT	跨房请求，加入和退出存在冲突
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_SRC_USERINFO_INVALID	跨房请求原用户信息无效
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_WITH_ORI	跨房房间与原用户房间相同
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_REPEAT	跨房请求房间重复
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_USER_EXISTED	跨房用户已存在
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_INVALID_REQUEST	无效请求
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_IS_NOT_EXIST	房间不存在
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_FRAME_TYPE_NOT_EQUAL	跨房源房间和目的房间加密模式不一致
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_AUTHENTICATION_FAILURE	鉴权失败
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_REMOVE_INFO_NOT_EXIST	退出跨房信息不存在
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_EXCEPTION_STOP	异常退出跨房

HWRtcRemoteMicState

表 6-124 麦克风设备状态

枚举值	描述
HWRTC_REMOTE_MIC_STATE_UNMUTE	麦克风设备状态正常
HWRTC_REMOTE_MIC_STATE_MUTE	麦克风设备状态静音

6.4.10 事件回调（HWRtcReplay）

本章节介绍了RtcEngineReplayKitExt SDK回调接口HWRtcReplayDelegate的详情。

replayBroadcastFinished

```
- (void)replayBroadcastFinished;;
```

【功能说明】

录屏结束回调。

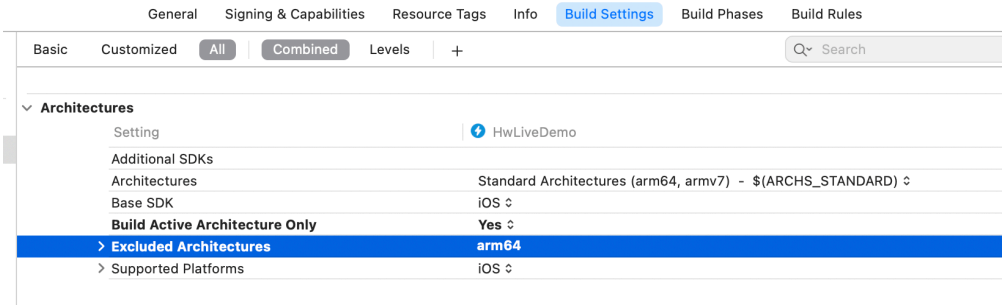
【回调参数】

无

6.5 常见问题

- **iOS平台如何缩减安装包体积?**

  对于iPhone 5s及以上版本的手机支持只打包arm64架构，可以在XCode的Build Setting < Build Active Architecture Only设置为YES，并将Valid Architectures只写arm64，则SparkRTC SDK可压缩一半的体积。

- **设置编码参数时，调用setVideoEncParam接口为什么会提示参数设置错误?**

  设置分辨率时需对照**华为SDK系统推荐的码表**才能设置成功。

- **加入房间失败时，如何解决?**

  首先通过返回的**客户端错误码**进行分析，主要有以下几个原因：

  - 网络问题，需确认网络是否正常运行。
  - 鉴权问题，应用默认开启鉴权，您需要确保鉴权生成正确，参数合理传入，且保证没过期，具体可参考**接入鉴权**。
  - 参数问题，确认关键参数是否填写，以及是否正确填写，比如username是必填的，countryCode无特殊需求，则填空字符串。countryCode值的填写具体请参见**国家码对照表**。

- **什么原因会导致跨房不成功?**

  可能性1：同一时间不同房间最多只有一个Joiner角色才能跨房成功。

  可能性2：同一时间最多只能跨4个房间，跨房对应的房间ID必须互不相同。

  可能性3：当前的远端音频模式（setRemoteAudioMode）设置为HWRtcRemoteAudioTopOfAll模式时不支持跨房。

- **怎么样使用远端音频模式为HWRtcRemoteAudioSubscribe时才能默认听不到远端用户的声音?**

  HWRtcRemoteAudioSubscribe为自主订阅，需要用户手动调用订阅。

方法1：在加入房间（joinRoom）时调用带有HRTCMediaOptions类的方法，创建该类实例后autoSubscribeAudio属性设置为NO，进入房间后听不到远端用户声音，需要手动调用muteRemoteAudio根据uid单个用户订阅才能听到声音。

方法2：在进入房间后调用setDefaultMuteAllRemoteAudioStreams接口设置是否自订阅远端用户音频流；当参数设置为false时，muteAllRemoteAudio不论参数是true还是false，新加入用户都不会订阅音频流。

注意：在调用setDefaultMuteAllRemoteAudioStreams接口之前，进入的远端用户不受该接口控制，重新进入后才受控制。

- **为什么onVideoStats、onAudioStatus、onSubStreamStats回调触发时程序崩溃?**

  回调函数的入参localStats和remoteStats指针有可能为空，需要先判断不为空再使用，否则可能引发空指针错误。

- **为什么本端听筒能听到自己的声音?**

  调用muteRemoteAudio时参数设置为自己uid就会发生此类情况。

- **setExternalAudioCapture（音频自采集）、setExternalVideoCapture（视频自采集）、setExternalMediaFrameOutput（视频自渲染）能在房间内开启吗?**

  不可以，需要在加入房间前调用，调用后在房间内不能修改。

- **iOS端是否可以监听远端离开房间?**

  可以使用onRemoteUserOffline监听用户离开房间事件。

- **一个房间里可以同时有多路屏幕分享吗?**

  不能，目前支持一个房间内只能有一路辅流屏幕分享。

- **为什么入会的时候没有声音?**

  可能性1：入会前没有订阅音频。

  可能性2：服务端出现问题。

  可能性3：远端没有开启视频流。

- **SparkRTC支持哪几种系统音量模式?**

  支持两种模式：媒体音量和通话音量。

  - 媒体音量：播放背景音乐、视频、混音的音量，媒体音量可以调整到零。
  - 通话音量：是指在进行音视频通话的音量，通话音量不可以调整到零。

6.6 修订记录

表 6-125 修订记录

修改时间	修改说明
2022-06-21	第十七次正式发布 本次变更如下： ● 新增以下接口： – addMultiRoomMediaRelay：添加单个跨房 – removeMultiRoomMediaRelay：删除单个跨房 – stopMultiRoomMediaRelay：停止所有跨房 – appendLocalView：设置本地视频另一个窗口显示 – appendRemoteView：设置远端视频另一个窗口显示 ● 新增以下事件回调： – onRemoteMicrophoneStateChanged：远端麦克风设备状态变更通知 – onUserNetworkQualityNotify：加入房间后的用户级网络质量状态回调 ● 新增以下数据类型： – HWRtcRemoteMicState：麦克风设备状态 – HRTCMultiRoomMediaRelayConfiguration：跨房配置 – HRTCSrcMultiRoomMediaInfo：源房间信息 – HRTCDstMultiRoomMediaInfo：目标房间信息 – HRTCMultiRoomMediaRelayState：跨房状态 – HRTCMultiRoomMediaRelayStateCode：跨房状态码
2022-03-24	第十六次正式发布 本次变更如下： 修改appid获取方式的相关描述。
2022-03-18	第十五次正式发布 本次变更如下： 新增setNetworkBandwidth接口

修改时间	修改说明
2022-02-25	第十四次正式发布 本次变更如下： 新增的接口及回调： • recordingDeviceTest音频采集设备测试（只支持macOS） • finishRecordingDeviceTest结束音频采集设备测试（只支持macOS） • playbackDeviceTest音频播放设备测试（只支持macOS） • finishPlaybackDeviceTest结束音频播放设备测试（只支持macOS）echoTest音频设备回路测试（只支持macOS） • finishEchoTest结束音频设备回路测试（只支持macOS） • cameraDeviceTest视频采集设备测试 • finishCameraDeviceTest结束视频采集设备测试 • onAudioDeviceTestVolumeNotify音频设备测试回调 • startScreenShareWithAppGroup开启屏幕共享(只支持iOS) • setupWithAppGroup屏幕录制启动。 • broadcastFinished屏幕录制关闭。 • sendVideoSampleBuffer发送屏幕录制数据。 • replayBroadcastFinished主动停止屏幕录制。
2021-12-02	第十三次正式发布 本次变更如下： 优化部分文档描述。
2021-11-22	第十二次正式发布 本次变更涉及部分API的逻辑优化与融合，如joinRoom、onConnectionChangedNotify、pushLocalVideo等，具体请参见**接口参考**。

修改时间	修改说明
2021-06-05	第十一次正式发布 本次发布版本为1.8.0版本，整合了之前若干版本和分支的一个全新版本，主要变更内容为跨房功能重构，结合服务端升级，可以支持2W人超大型会议和最多同时跨5个房间互动等场景。此外，管控面和用户体验上也新增了部分功能和优化。 本次变更如下： 新增的接口及回调： ● **changeUserName**、renewAuthorization、onUserNameChangedNotify：会议中修改用户昵称的接口、签名更新的接口、修改用户昵称的回调。 ● onFirstRemoteAuxiliaryStreamDecoded：引擎收到第一帧远端辅流并解码成功的回调。 ● **createConnection**：加入多房间（跨房）前，与要跨入的房间先建立连接的接口，跨一个房间建立一个连接。 ● setRemoteAudioMode：设置音频订阅模式的接口，整合了setRemoteAudioTopNVoice和enableTopThreeAudioMode两个接口。 废弃的接口及回调： ● connectOtherRoom、onConnectOtherRoom、disconnectOtherRoom、onDisconnectOtherRoom：新跨房通过与对应房间先建立连接HWRtcConnection，再通过连接调用其下的joinRoom和leaveRoom接口实现跨房和退房功能，回调亦通过HWRtcConnection下属回调OnJoinRoomSuccess/Failure、OnLeaveRoom实现，具体请参见**3.3.3HWRtcConnection** 和**3.3.4事件回调（HWRtcConnection）**章节，原HWRtcEngine下属相关接口和回调已废弃。 ● setRemoteAudioTopNVoice、enableTopThreeAudioMode：经setRemoteAudioMode整合后已废弃。 参数变更的接口及回调： changeUserRole：不再需要roomId参数，跨房所在房间用HWRtcConnection连接调用角色切换接口，HWRtcEngine只作其所在房间的角色切换。
2021-01-28	第十次正式发布 本次变更如下： ● **HWRtcEngine**新增接口：getVersion

修改时间	修改说明
2020-12-25	第九次正式发布 本次变更如下： • **HWRtcEngine**新增接口：pullRemoteVideo、pullAllRemoteVideo、enableSmallVideoStream、setPriorRemoteVideoStreamType、setRemoteVideoStreamType、setupRemoteView、setRemoteVideoAdjustResolution、setVolumeNotifyInterval • 新增**事件回调**：onUserAuxiliaryStreamAvailable • 新增**数据类型**：HWRtcVideoStreamType、HWRtcVideoEncodeResolutionMode • 更新客户端错误码
2020-12-17	第八次正式发布 本次变更如下： • 删除接口onStreamAvailable • 新增接口onAuxiliaryStreamStatsNotify • 本地预览画面接口修改为setupLocalVideo • joinRoom回调与Windows端保持一致
2020-11-26	第七次正式发布 本次变更如下： 新增SparkRTC**接入鉴权**方法说明
2020-10-21	第六次正式发布 本次变更如下： • **HWRtcEngine**新增接口：setPlaybackDevices、setRecordingDevices、getRecordDevices、startAudioFile、stopAudioFile、pauseAudioFile、resumeAudioFile、setExternalAudioCapture、setExternalVideoCapture、pushExternalAudioFrame、pushExternalVideoFrame、setExternalMediaFrameOutput • 新增**事件回调**：onUserVolumeStatsNotify、onStartAudioFile、onStopAudioFile、onPauseAudioFile、onResumeAudioFile、onRenderExternalVideoFrame、onPlaybackExternalAudioFrame • 新增**数据类型**：HWRTCMediaDirectionHWRtcAudioFilePlayMode、HWRtcAudioFrameType、HWRtcVideoImageFormat、HWRtcAudioFrame、HWRtcVideoFrame、HWRtcStartAudioFileParam

修改时间	修改说明
2020-09-04	第五次正式发布 本次变更如下： ● 增加视频镜像接口：，setVideoEncoderMirror ● 增加音视频流采集发流控制接口及流状态变化回调：enableLocalVideo，onRemoteAudioStateChangedNotify，onRemoteVideoStateChangedNotify ● HRtcErrorCode增加90000040~90000042错误码 ● 增加HRTCLeaveReason，HRTCVideoMirrorType，HRTCRemoteAudioStreamState，HRTCRemoteAudioStreamStateReason，HRTCRemoteVideoStreamState，HRTCRemoteVideoStreamStateReason枚举
2020-08-17	第四次正式发布 本次变更如下： ● 增加选看跨房功能接口及回调：connectOtherRoom，disconnectOtherRoom，onConnectOtherRoom，onDisconnectOtherRoom ● 增加音频流接收选择接口：muteRemoteAudio，muteAllRemoteAudio ● 增加音视频流统计信息上报：onVideoStatsNotify，onAudioStatsNotify，onAuxiliaryStreamStatsNotify ● 增加卡顿统计信息上报：onLeaveRoom(HRTCStatsInfo) ● HRtcErrorCode增加90000034~90000039错误码 ● 服务端错误码增加RTC.32000030~RTC.32000033错误码 ● 优化升级HRTCStatsInfo类
2020-07-03	第三次正式发布 本次变更如下： ● 选看远端接口修改为startRemoteStreamView，updateRemoteRenderMode，stopRemoteStreamView ● 增加辅流相关接口startRemoteAuxiliaryStreamView，stopRemoteAuxiliaryStreamView，setRemoteAuxiliaryStreamViewRotation，updateRemoteAuxiliaryStreamRenderMode ● 修改错误码等

修改时间	修改说明
2020-06-20	第二次正式发布 本次变更如下： ● SDk集成中，修改需要添加的库文件。 ● 修改接口参考中的类、方法等内容，具体如下所示： – RtcEngine类增加setVideoEncoder、changeUserRole、adjustRecordingVolume等方法。 – HwRtcEngineDelegate类增加rtcEngineUserRoleChangeWithNewRole等方法。 – 增加HwRtcVideoEncode、HwRtcVideoStatsInfo和HwRtcEncodeStreamType类。 – HwRtcStreamType枚举类增加类成员。 – RtcErrorCode增加90000019~90000030的错误码。
2020-04-15	第一次正式发布

7 All Platform C++ SDK

7.1 开发前准备

7.1.1 Android

前提条件

已**提交工单**获取SDK包。

环境要求

OHOS SDK需要集成到APP工程中，建议您在如下推荐环境中进行集成开发。

- 准备DevEco，推荐使用4.0.0(10) SDK及以上。
- 准备Android运行环境：API 21、Android 5.0以上设备。
- 支持的终端CPU架构：armeabi-v7a、arm64-v8a。

说明

手机的CPU架构可通过以下方式查询。

手机开启USB调试，连接上电脑，然后打开Windows操作系统中的cmd程序，输入如下命令：

```
adb shell getprop ro.product.cpu.abi
```

SDK 集成

步骤1 解压Android SDK包。

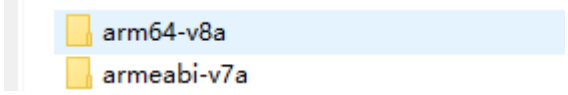
步骤2 将Android SDK包中的“hwRtcSdk.aar”等aar文件，导入Android Studio工程的libs文件夹下。

步骤3 在“/app/build.gradle”文件中设置依赖本地aar。

```
// 依赖本地aar
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar', '*.aar'])
}
```


步骤4 将如图7-1所示的包含so库的两个文件夹导入到jniLibs文件夹中。

图 7-1 so 库目录



步骤5 在“/app/build.gradle”文件中设置so库的存放路径。

```
sourceSets {
    main {
        jniLibs.srcDirs = ['src/main/jniLibs']
    }
}
```

步骤6 在“app/src/main/res/values/strings.xml”文件中配置appId。其中，appId请参考应用管理获取。

```
<string name="setting_appId_title" translatable="false">appId</string>
```

步骤7 在“/app/src/main/AndroidManifest.xml”文件中配置App权限。

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

步骤8 单击“Sync Project With Gradle Files”，同步项目文件，完成SDK集成。

----结束

7.1.2 iOS

前提条件

已提交工单获取SDK包。

环境要求

- 准备XCODE集成开发。
- 准备iOS 8.0及以上的iPhone真机。
- 支持的终端CPU架构：arm64，arm32。

SDK 集成

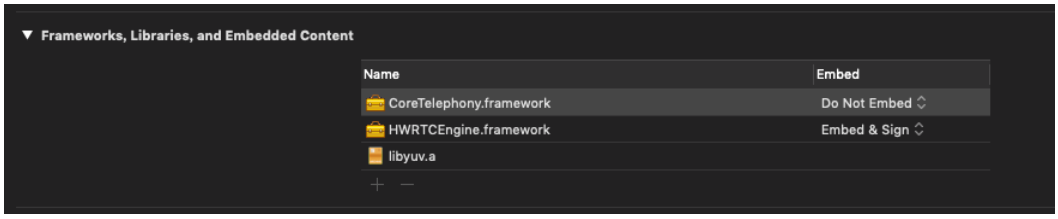
步骤1 解压iOS SDK包。

步骤2 将iOS SDK包中的HWRTCEngine动态库和RTC.bundle文件导入创建的XCODE工程中。

图 7-2 导入 HWRTCEngine 动态库和 RTC.bundle 文件

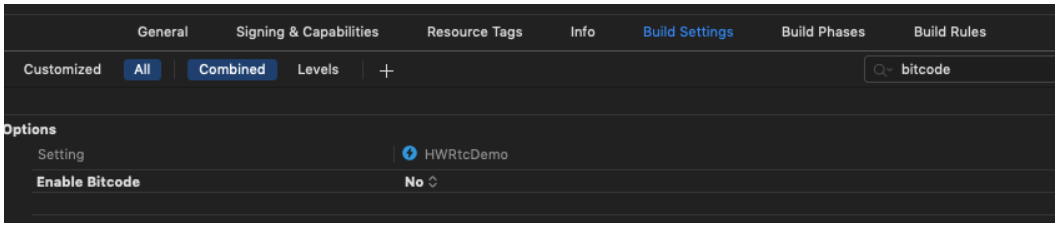
步骤3 在“General”页面将SDK中HWRTCEngine.framework文件加入到工程。如果需要使用混音功能，请添加hwffmpeg.framework库文件至HWRTCEngine.framework同级目录。

图 7-3 添加 HWRTCEngine.framework



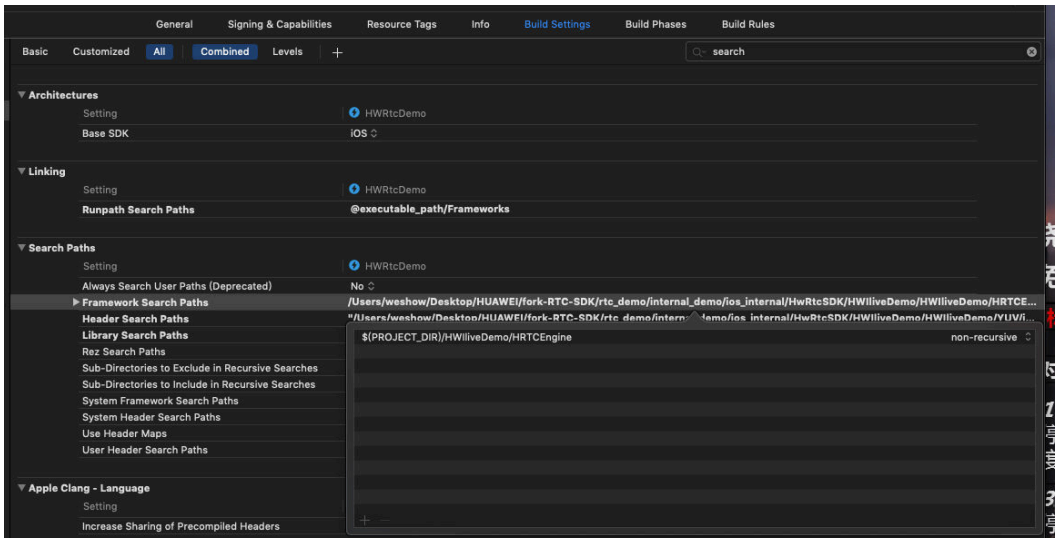
步骤4 在“Build Settings”页面关闭“Enable Bitcode”， 将其设置为“No”。

图 7-4 设置 Enable Bitcode



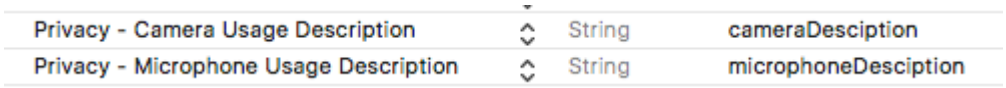
步骤5 在“Build Settings”页面的搜索框输入“search”，查看Framework search paths路径是否正确，确保文件加载成功。

图 7-5 检查文件是否加载成功



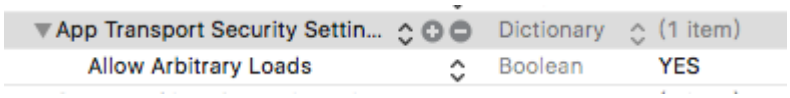
步骤6 在“info.plist”文件中增加摄像头和麦克风权限。

图 7-6 摄像头和麦克风权限



步骤7 在“info.plist”文件中添加ATS。

图 7-7 添加 ATS



步骤8 使用XCODE连接iPhone，编译工程，若界面提示“Build Success”，则完成SDK集成。

----结束

7.1.3 Mac

前提条件

已提交工单获取SDK包。

环境要求

- 准备XCODE集成开发环境。
- 准备MAC设备，支持macOS 10.11以上的设备。
- 支持的终端CPU架构：x86_64。

SDK 集成

支持dylib和framework两种包集成。

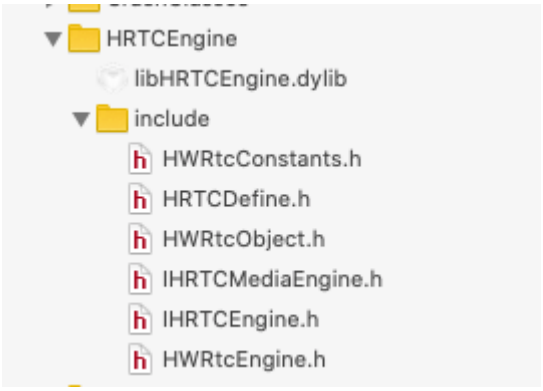
步骤1 解压Mac SDK包。

步骤2 将MAC SDK包中的lib动态库libHRTCEngine.dylib或者libHRTCEngine.framework和RTC.bundle文件导入创建的XCODE工程中，如果用到混音功能，需要将hwffmpeg.framework至libHWRTCEngine.framework同级目录。

图 7-8 lib 库

步骤3 将MAC SDK包中的头文件“HWRtcObject.h”、“HWRtcConstants.h”、“HRTCDefine.h”“IHRTCMediaEngine.h”、“IHRTCEngine.h”和“HWRtcEngine.h”导入工程中，集成framework在Headers下面自带头文件，无需再导入。

图 7-9 导入头文件



步骤4 在“Build Settings”页面的搜索框输入“search”，确保头文件和库文件的位置都已经在XCODE设置成功。

若“Header Search Paths ”和“Library Search Paths”中文件位置不对，可以将XCODE对应文件夹直接拖过来即可。

图 7-10 检查文件是否加载成功

步骤5 编译工程，若界面提示“Build Success”，则完成SDK集成。

----结束

7.1.4 Windows

前提条件

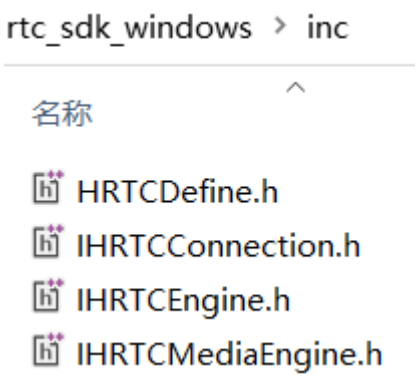
已**提交工单**获取SDK包。

环境要求

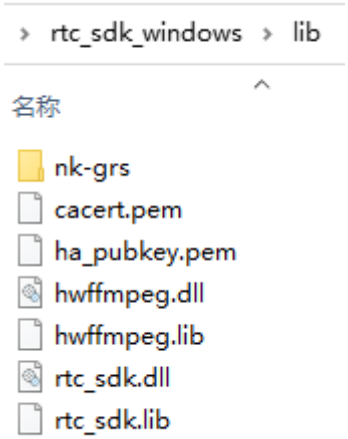
- 准备集成开发环境，建议使用Microsoft Visual Studio 2015或以上版本。
- 支持Windows 7或以上版本的Windows设备。
- 支持的平台：x86 release。

资源文件列表

- 头文件目录



- lib文件目录



SDK 集成

步骤1 用Visual Studio打开示例Demo或您的客户端工程文件，解压Windows SDK包，获取资源文件列表中所示的文件目录。

步骤2 在“配置属性 > VC++目录 > 包含目录”中配置“包含目录”值。

```
..\..\..\rtc_sdk_windows\inc;
```

步骤3 在“配置属性 > 链接器 > 常规 > 附加包含目录”中添加“lib文件目录”。

```
..\..\..\rtc_sdk_windows\lib;
```

步骤4 在“配置属性 > 链接器 > 输入 > 附加依赖项”中添加“lib文件”。

```
rtc_sdk.lib
```

步骤5 在“生成事件 > 预生成事件 > 命令行”添加动态库文件到输出路径(包含nk-grs目录)。

```
xcopy /Y ..\..\..\rtc_sdk_windows\release\dll\* $(Outdir) /s /e
```

步骤6 执行编译，完成SDK集成。

----结束

7.2 SDK 使用

以下为集成SDK进行实时音视频互动直播的基本流程：

步骤1 创建引擎。

appId获取方法请参见**创建应用**。

```
m_pRtcEngine = createHRtcEngine(); // 创建引擎
HRTCLogConfig logConfig;
logConfig.level = HRTC_LOG_LEVEL_DEBUG;
logConfig.path = logPath;         // logPath为目录，非文件
logConfig.logSize = 10 * 1024;
logConfig.enable = true;          // 开启日志记录

HRTCEngineConfig engineConfig;
engineConfig.appId = appId;   // appId需在控制台中创建应用后获取.
engineConfig.countryCode = countryCode; // 可以根据Grs国家码对照表传值，建议传"CN"
engineConfig.enableHaTrace = true; // 打点开关

HRTCEngineContext engineContext;
engineContext.logConfig = logConfig;
engineContext.engineConfig = engineConfig;
engineContext.eventHandler = &m_engineEventHandler; // 事件回调对象指针
m_pRtcEngine->initialize(&engineContext); // 初始化引擎
安卓平台调用m_pRtcEngine->initialize(engineInfo, eventHandler);
```

- **m_engineEventHandler**：引擎回调句柄，指定一个回调事件。SDK通过指定的事件通知应用程序的运行事件，如加入或离开房间等，具体请参见**事件回调（IHRTCEngine）**。
- **engineInfo**：结构体具体请参见**HRTCEngineConfig**。
- **context**：引擎初始化参数，具体请参见**HRTCEngineContext**。

步骤2 加入房间。

```
HRTCJoinParam joinRoomParam;
memset(&joinRoomParam, 0, sizeof(HRTCJoinParam));
joinRoomParam.autoSubscribeAudio = true;
joinRoomParam.autoSubscribeVideo = false;

joinRoomParam.userId = userId; // userId用于标识同一房间的不同用户
joinRoomParam.userName = userName; // 用户昵称，如无特殊需求，保持和userId一致即可
joinRoomParam.authorization = authorization;
joinRoomParam.ctime = m_ctime;
joinRoomParam.roomId = roomId;
joinRoomParam.userRole = m_roleType;
joinRoomParam.scenario = 1;
m_pRtcEngine->joinRoom(joinRoomParam);
```

- **m_roleType**：用户角色。
- **userId**：本端用户唯一标识。
- **userName**：用户昵称，该昵称为UTF-8编码。
- **roomId**：房间ID，房间唯一标识。
- **scenario**：使用的场景（0=主动订阅（默认）；1=TopN（千人）；2=P2P; 3=RTSA CMD自动订阅）。

- **ctime**：当前时间戳。
- **authorization**：签名认证。签名的具体生成方法请参见**接入鉴权**，支持最大长度为1024。
- **autoSubscribeAudio**：是否主动订阅音频。
- **autoSubscribeVideo**：是否主动订阅视频。

步骤3 设置本地视图。

```
m_pRtcEngine->setupLocalView(hwnd, HRTC_VIDEO_DISPLAY_MODE_HIDDEN);
```

- **hwnd**：视频窗口句柄。
- **HRTC_VIDEO_DISPLAY_MODE_HIDDEN**：图像填充模式，具体请参见**HRTCVideoDisplayMode**。

步骤4 当连麦者加入房间，设置远端窗口。

```
void HWEngineEventHandler::onRemoteUserOnline(const char* roomId, const char* userId, const char*
nickName)
{
    wchar_t *room = StringUtility::HW_Utf8ToUnicodeW(roomId);
    wchar_t *user = StringUtility::HW_Utf8ToUnicodeW(userId);
    wchar_t *name = StringUtility::HW_Utf8ToUnicodeW(nickName);
    UserJoinInfo* userInfo = new UserJoinInfo();
    _tcscpy(userInfo->roomId, room);
    _tcscpy(userInfo->userId, user);
    _tcscpy(userInfo->nickname, name);
    if(m_hMainWnd != NULL) {
        ::PostMessage(m_hMainWnd, WM_USERJOIN, (WPARAM)userInfo, 0);
    }
    if (room != NULL) {
        free(room);
    }
    if (user != NULL) {
        free(user);
    }
    if (name != NULL) {
        free(name);
    }
}

//远端用户加入房间触发onRemoteUserOnline回调，设置远端窗口。
if (m_pRtcEngine->startRemoteStreamView(userId, m_wndVideo[nIndex].GetSafeHwnd(), streamType) ==
0) {
    m_pRtcEngine->updateRemoteRenderMode(userId,
HRTC_VIDEO_DISPLAY_MODE_HIDDEN,HRTC_VIDEO_MIRROR_TYPE_DISABLE);
}
```

- **userId**：远端用户id，唯一标识。
- **m_wndVideo[nIndex].GetSafeHwnd()**：视频窗口句柄。
- **streamType**：视频分辨率，支持全高清、高清、标清和流畅选择。
- **HRTC_VIDEO_DISPLAY_MODE_HIDDEN**：图像填充模式，具体请参见**HRTCVideoDisplayMode**。

步骤5 离开房间。

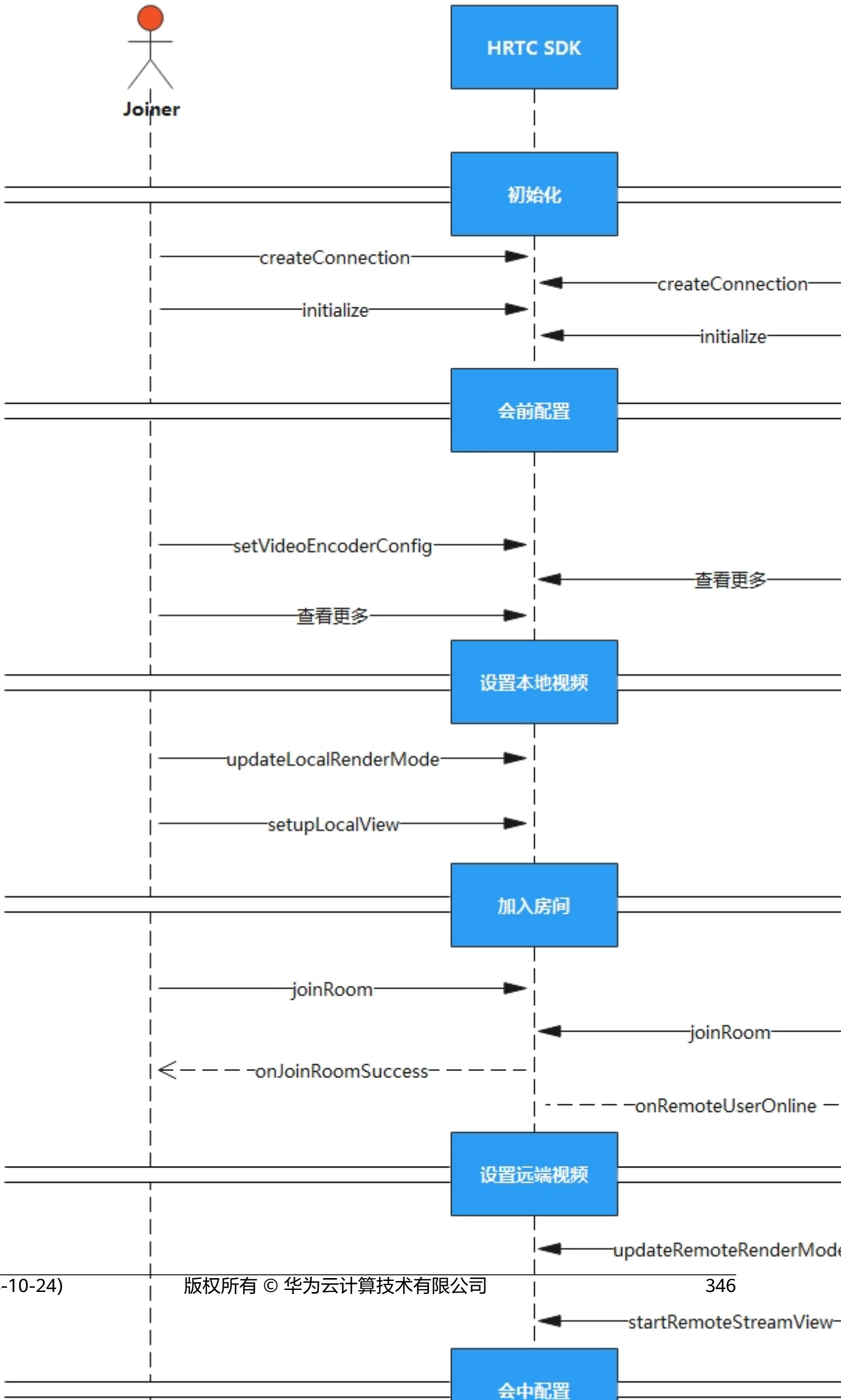
```
m_pRtcEngine->leaveRoom();
```

根据场景需要，如结束通话、关闭App或App切换至后台时，调用leaveRoom离开当前通话房间。

至此，互动直播基本流程可以成功运行。

----结束

7.3 基本使用逻辑



1. 创建新的项目工程，导入SDK后，需要先创建引擎。
2. 您可以在入会前进行视频编码、声音播放模式等参数的配置。
3. 设置本地视图。
4. 加入房间后，将通过回调的方式通知房间内的其他用户，收到用户加入的回调后，可以为其设置远端视图。
5. 在会中，也可以进行切换摄像头等参数的配置。
6. 离开房间后，需销毁对应资源。

说明

在时序图中，单击相应接口名称可快速跳转到相应接口位置查看其使用方法。

7.4 接口参考

7.4.1 IHRTCEngine

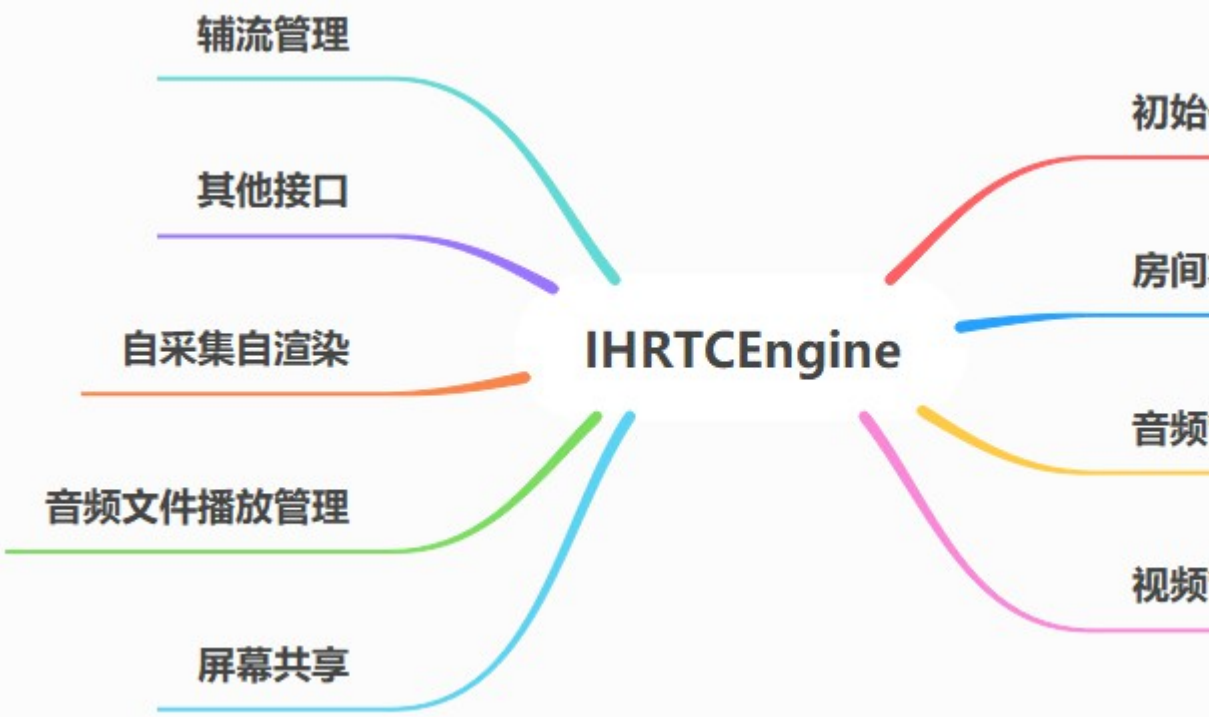
7.4.1.1 接口总览

本章节介绍了全平台C++ SDK的IHRTCEngine接口详情。

IHRTCEngine按照其功能可分类为：初始化等基础接口、房间功能、视频管理、辅流管理、屏幕共享、音频管理、音效文件播放管理、自采集自渲染、其他接口。

说明

单击下图中相应接口名称，可快速跳转到相应接口位置查看其使用方法。



7.4.1.2 接口按功能说明

初始化等基础接口

表 7-1 初始化等基础接口

接口	描述	windows是否支持	Mac是否支持	iOS是否支持	Android是否支持
createHRtcEngine	创建IHRTCEngine对象	√	√	√	√
getHRtcEngine	获取创建后的IHRTCEngine对象	√	√	√	√
enableStats	开启打点统计，在initialize前调用	√	√	√	√
initialize	IHRtcEngine对象初始化函数	√	√	√	√
release	释放IHRtcEngine对象资源，退出时调用	√	√	√	√
logUpload	开启日志上传	√	√	√	√
getVersion	获取当前SDK版本号	√	√	√	√
getAudioDeviceManager	获取系统音频设备管理对象	√	√	√	√
getVideoDeviceManager	获取系统视频设备管理对象	√	√	√	√
setJniLoadParams	设置jvm context 仅安卓使用	×	×	×	√
setEncryption	设置端到端加密模式	√	√	√	√
setAccessResourceType	设置接入环境	√	√	√	√
setNetworkBandwidth	设置网络带宽限制	√	√	√	√
renewAppid	设置Appid	√	√	√	√

房间功能

表 7-2 房间功能接口

接口	描述	windows是否支持	Mac是否支持	iOS是否支持	Android是否支持
joinRoom	加入房间	√	√	√	√
leaveRoom	离开房间	√	√	√	√
renewAuthorization	签名更新	√	√	√	√
changeUserRole	设置用户的角色，切换角色时使用	√	√	√	√
changeUserName	设置用户自己的昵称	√	√	√	√
createConnection	创建连接。跨房前需先创建连接	√	√	√	√
addMultiRoomMediaRelay	添加单个跨房	√	√	√	√
removeMultiRoomMediaRelay	删除单个跨房	√	√	√	√
stopMultiRoomMediaRelay	停止所有跨房	√	√	√	√
subscribeAISubtitles	订阅AI字幕	√	√	√	√

视频管理

表 7-3 视频管理接口

接口	描述	windows是否支持	Mac是否支持	iOS是否支持	Android是否支持
enableLocalVideo	设置是否开启本地摄像头采集视频	√	√	√	√
setVideoEncoderConfig	设置视频发流编码参数	√	√	√	√

接口	描述	windows是否支持	Mac是否支持	iOS是否支持	Android是否支持
enableSmallVideoStream	大小流模式设置是否开启小流并设置小流编码参数	√	√	√	√
startLocalPreview	开始本地预览	√	√	√	√
stopLocalPreview	关闭本地预览	√	√	√	√
setupLocalView	设置本地渲染视图	√	√	√	√
updateLocalRenderMode	设置本地视图渲染模式，镜像模式	√	√	√	√
setupRemoteView	设置远端流渲染视图（发起选看，参数置空则停止选看）	√	√	√	√
updateRemoteRenderMode	设置远端用户视图渲染模式，镜像模式	√	√	√	√
setRemoteVideoStreamType	大小流模式，设置远端视频流类型	√	√	√	√
setPriorRemoteVideoStreamType	大小流模式，设置所有订阅的远端视频流类型。	√	√	√	√
pushLocalVideo	设置是否发送本地视频流	√	√	√	√
pullAllRemoteVideo	设置是否接收所有用户的视频流	√	√	√	√
startRemoteStreamView	设置远端用户渲染视图（发起选看-老接口）	√	√	√	√
stopRemoteStreamView	关闭远端用户的渲染视图（停止选看）	√	√	√	√
setRemoteVideoAdjustResolution	设置是否开启远端分辨率自适应	√	√	√	√
setVideoEncoderMirror	设置编码器输出的画面（本地发流）镜像模式	√	√	√	√
setCameraConfig	设置摄像头参数	×	×	√	√
switchCamera	切换摄像头，移动端	×	x	√	√
startAllRemoteView	批量设置远端流视图	√	√	√	√

接口	描述	windows是否支持	Mac是否支持	iOS是否支持	Android是否支持
setRemoteViewRotation	设置远端流视图旋转角度	×	x	√	√
setRemoteViewOrientation	设置远端流视图方向(横竖屏)	×	x	√	√
setVideoPaddingImage	设置关闭视频发流时发送的图片	√	√	√	√
pullAllRemoteVideo	设置是否接收所用户的视频流	√	√	√	√
setBeautyRetouchOption	设置是否开启美颜功能	×	x	√	√
enableVideoBrighten	设置是否开启增亮功能	√	×	x	x
ApplyGSensorMode	设置是否开启重力感应	×	x	√	x
startPublishStream	开始旁路推流	√	√	√	√
updateTransCoding	更新旁路推流	√	√	√	√
stopPublishStream	停止旁路推流	√	√	√	√
setVideoWaterMark	插入/删除水印	√	x	x	x
setBackgroundBlur	设置本地视频背景虚化	√	x	x	√
setBackgroundReplace	设置本地视频背景替换	√	x	x	√
appendLocalView	设置本地视频另一个窗口显示	√	√	x	x
appendRemoteView	设置远端视频另一个窗口显示	√	√	x	x

辅流管理

表 7-4 辅流管理接口

接口	描述	windows是否支持	Mac是否支持	iOS是否支持	Android是否支持
setRemoteAuxiliaryStreamViewRotation	设置远端辅流视图旋转角度	×	x	√	√
startRemoteAuxiliaryStreamView	开启辅流渲染视图（发起辅流选看）	√	√	x	√
stopRemoteAuxiliaryStreamView	关闭辅流渲染视图（停止辅流选看）	√	√	x	√
updateRemoteAuxiliaryStreamRenderMode	设置辅流视图显示模式，镜像模式	√	√	x	√
setRemoteAuxiliaryStreamViewOrientation	设置远端辅流视图方向（横竖屏）	×	x	√	√
setAuxiliaryVideoEncoderConfig	设置辅流编码参数	√	√	x	√
setAuxiliaryVideoEncodeSmooth	设置是否开启辅流的流畅度优先（降低辅流选看分辨率）	√	√	x	√

屏幕共享

表 7-5 屏幕共享接口

接口	描述	windows是否支持	Mac是否支持	iOS是否支持	Android是否支持
getScreenShareSources	获取屏幕可共享对象列表	√	√	x	√
setScreenShareTarget	选择屏幕共享对象	√	√	x	√
startScreenShare	开启屏幕共享	√	√	x	√
stopScreenShare	停止屏幕共享	√	√	x	√
addHiddenShareWindow	将指定窗口加入屏幕共享排除列表	√	√	x	√

接口	描述	windows是否支持	Mac是否支持	iOS是否支持	Android是否支持
deleteHiddenShareWindow	将指定窗口从屏幕共享排除列表中移除	√	√	x	√
removeAllHiddenShareWindow	将所有窗口从屏幕共享排除列表中移除	√	√	x	√
getScreenShareSources	获取屏幕可共享对象列表，包含缩略图	√	x	x	x

音频管理

表 7-6 音频管理接口

接口	描述	windows是否支持	Mac是否支持	iOS是否支持	Android是否支持
enableLocalAudioStream	设置是否开启麦克风音频采集	√	√	√	√
adjustRecordingVolume	设置麦克风采集的音量（0-100）	√	√	√	√
adjustPlaybackVolume	调整扬声器播放的音量（0-100）	√	√	√	√
setShareComputerSound	设置是否开启系统音频采集、发送	√	√	x	x
muteLocalAudio	设置是否发送本地音频流	√	√	√	√
muteRemoteAudio	设置是否接收指定远端用户的音频流	√	√	√	√
muteAllRemoteAudio	设置是否接收远端所有用户的音频流	√	√	√	√
enableUserVolumeNotify	设置开启/关闭所有用户音量值上报并设置上报周期	√	√	√	√
setSpeakerModel	设置声音播放模式	x	x	√	√
echoTest	开始语音通话回路测试	√	x	x	x
finishEchoTest	停止语音直播回路测试	√	x	x	x
setDefaultSpeakerModel	设置默认声音播放模式	x	x	√	√

接口	描述	windows是否支持	Mac是否支持	iOS是否支持	Android是否支持
enableAudioTNR	设置是否开启音频降噪功能	√	√	√	√
sendAudioSeiMsg	发送音频SEI消息	√	√	√	√
setAudioFrameRecordParameters	设置音频采集回调的参数	√	√	√	√

音频文件播放管理

表 7-7 音频文件播放管理接口

接口	描述	windows是否支持	Mac是否支持	iOS是否支持	Android是否支持
startAudioFile	播放音频文件	√	√	√	√
stopAudioFile	停止播放音频文件	√	√	√	√
pauseAudioFile	暂停播放音频文件	√	√	√	√
resumeAudioFile	恢复播放音频文件	√	√	√	√
adjustAudioFileVolume	调整音频音量	√	√	√	√
adjustAudioFilePlayoutVolume	调整本地播放音频音量	√	√	√	√
getAudioFileVolume	获取音频音量	√	√	√	√
getAudioFilePlayoutVolume	获取本地播放音频音量	√	√	√	√
getAudioFileDuration	获取音频时长	√	√	√	√
getAudioFilePosition	获取音频播放位置	√	√	√	√
setAudioFilePosition	设置音频播放位置	√	√	√	√
getAudioClipsVolume	获取音效总音量	√	√	√	√

接口	描述	windows是否支持	Mac是否支持	iOS是否支持	Android是否支持
setAudioClipsVolume	设置音效总音量	√	√	√	√
getVolumeOfAudioClip	获取指定音效音量	√	√	√	√
setVolumeOfAudioClip	设置指定音效音量	√	√	√	√
playAudioClip	播放音效文件	√	√	√	√
stopAudioClip	停止播放指定音效	√	√	√	√
stopAllAudioClips	停止播放所有音效	√	√	√	√
pauseAudioClip	暂停播放指定音效	√	√	√	√
pauseAllAudioClips	暂停播放所有音效	√	√	√	√
resumeAudioClip	恢复播放指定音效	√	√	√	√
resumeAllAudioClips	恢复播放所有音效	√	√	√	√
getAudioClipCurrentPosition	获取指定音效当前播放位置	√	√	√	√
setAudioClipPosition	设置指定音效播放位置	√	√	√	√
getAudioClipDuration	获取音效文件时长	√	√	√	√

自采集自渲染

表 7-8 自采集自渲染接口

接口	描述	windows是否支持	Mac是否支持	iOS是否支持	Android是否支持
setExternalVideoCapture	设置是否开启外部视频自采集	√	√	√	√

接口	描述	windows是否支持	Mac是否支持	iOS是否支持	Android是否支持
pushExternalVideoFrame	推送外部视频数据	√	√	√	√
setExternalVideoFrameOutput	设置是否开启视频流自渲染	√	√	√	√
setAuxiliaryExternalVideoFrameOutput	设置是否开启辅流自渲染	√	√	√	√
setExternalAudioCapture	设置是否开启外部音频自采集	√	√	√	√
pushExternalAudioFrame	推送外部音频数据	√	√	√	√
setExternalAudioFrameOutput	设置是否开启音频流自渲染	√	√	√	√

其他接口

表 7-9 其他接口

接口	描述	windows是否支持	Mac是否支持	iOS是否支持	Android是否支持
startNetworkTest	开启会前网络质量测试	√	√	√	√
stopNetworkTest	停止会前网络质量测试	√	√	√	√
setAudioConfig	设置使用场景	√	√	√	√

7.4.1.3 初始化等基础接口

createHRtcEngine

```
huawei::rtc::IHRTCEngine* createHRtcEngine(void);
```

【功能说明】

创建IHRTCEngine对象。

【返回参数】

- huawei::rtc::IHRTCEngine*：IHRTCEngine对象。
- NULL：创建失败。

getHRtcEngine

```
huawei::rtc::IHRTCEngine* getHRtcEngine(void);
```

【功能说明】

调用createHRtcEngine成功后，可通过该函数，获取IHRTCEngine对象。

【返回参数】

- huawei::rtc::IHRTCEngine*：IHRTCEngine对象。
- NULL：IHRTCEngine对象不存在时返回。

enableStats

```
virtual int enableStats(bool enabled)
```

【功能说明】

开启打点统计，在initialize前调用。

【请求参数】

enabled：是否开启打点，默认开启。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

initialize

virtual int initialize(const HRTCEngineContext &context);

【功能说明】

IHRtcEngine对象初始化函数。

【请求参数】

context：引擎初始化参数, 具体请参见**HRTCEngineContext**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

release

```
virtual void release()
```

【功能说明】

释放IHRtcEngine对象资源。

⚠ 注意

如果资源已经释放，需要重新调用**createHRtcEngine**和**initialize**进行初始化。

logUpload

```
virtual int logUpload()
```

【功能说明】

开启日志上传。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 加入房间后才能主动上传日志。
- 会触发以下回调：
  - **onLogUploadResult**：日志上传结果回调。
  - **onLogUploadProgress**：日志上传进度回调，远端服务器不通时不会收到此回调。

getVersion

```
virtual char* getVersion()
```

【功能说明】

获取当前SDK版本号。

【请求参数】

无

【返回参数】

SDK当前版本号。

getAudioDeviceManager

```
virtual IHRTCAudioDeviceManager* getAudioDeviceManager()
```

【功能说明】

获取系统音频设备管理对象。通过该对象进行音频设备管理，具体请参见**音频设备管理**。

【请求参数】

无

【返回参数】

- IHRTCAudioDeviceManager*：IHRTCAudioDeviceManager对象。
- NULL：返回失败。

getVideoDeviceManager

```
virtual IHRTCVideoDeviceManager* getVideoDeviceManager()
```

【功能说明】

获取系统视频设备管理对象。通过该对象进行视频设备管理，具体请参见**视频设备管理**。

【请求参数】

无

【返回参数】

- IHRTCVideoDeviceManager*：IHRTCVideoDeviceManager对象。
- NULL：返回失败。

setJniLoadParams

```
int setJniLoadParams(void *jvm, void *context);
```

【功能说明】

安卓全平台传递jvm context参数，需要在**initialize**之前设置。

setEncryption

```
virtual int setEncryption(const HRTCEncryptionConfig &cryption)
```

【功能说明】

设置端到端加密模式，必须调用接口，在加入房间前调用生效。

【请求参数】

cryption：加密参数，具体请参见**HRTCEncryptionConfig**。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HRTCErrorCode**。

setAccessResourceType

```
virtual int setAccessResourceType(int resType) = 0;
```

【功能说明】

设置接入的环境，不支持跨房间场景。

【请求参数】

resType：环境类型。

- 0：公网sfu资源。
- 1：公司局Sfu。

【返回参数】

- 0：方法调用成功。
- <0：方法调用失败。具体请参见**HRTCErrorCode**。

setAccessResourceType

```
virtual int setAccessResourceType(const char
*resourceTags[HRTCConstant::HRTC_MAX_RESOURCE_TAG_COUNT], int num, int intranetType) = 0;
```

【功能说明】

指定使用的sfu服务器，设置接入的环境，不支持跨房间场景。

【请求参数】

resourceTags：资源标签，指定连接的sfu

num：资源标签数组长度

intranetType：环境类型。

- 0：公网sfu资源。
- 1：公司局Sfu。

【返回参数】

- 0：方法调用成功。
- <0：方法调用失败。具体请参见**HRTCErrorCode**。

setNetworkBandwidth

```
virtual int setNetworkBandwidth(const HRTCNetworkBandwidth &bandwidthParam)
```

【功能说明】

设置网络带宽限制，在每次加入房间之前调用。

【请求参数】

bandwidthParam：带宽设置参数，具体请参见**HRTCNetworkBandwidth**。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HRTCErrorCode**。

renewAppid

```
virtual int renewAppid(const char* appid)
```

【功能说明】

设置Appid，用来更新AppId，在加入房间调用前才会生效，否则只能下一次入会生效。

【请求参数】

appid：设置appid。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HRTCErrorCode**。

7.4.1.4 房间功能

joinRoom

```
virtual int joinRoom(const HRTCJoinParam &joinParam)
```

【功能说明】

加入房间。该方法让用户加入通话房间。

【请求参数】

joinParam：加入房间信息。具体请参见**HRTCJoinParam**。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

会触发以下回调：

- **onConnectStateChange**：连接状态发送改变。
- **onJoinRoomSuccess**：加入房间成功。
- **onJoinRoomFailure**：加入房间失败。
- **onRemoteUserOnline**：远端用户收到当前用户加入房间的通知。

音频的自动订阅策略设置只在音频订阅模式下生效。

leaveRoom

```
virtual int leaveRoom()
```

【功能说明】

离开房间。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

【注意事项】

⚠ 注意

会触发以下回调：

- **onLeaveRoom**：离开房间回调。
- **onConnectStateChange**：连接状态改变回调。
- **onRemoteUserOffline**：远端用户收到当前用户离开房间的通知。

renewAuthorization

```
virtual int renewAuthorization(const char* signature, long long ctime)
```

【功能说明】

鉴权签名过期，收到**onAuthorizationExpired**签名鉴权过期回调后，更新鉴权签名。

【请求参数】

- signature：鉴权签名字符串。
- ctime：过期时间。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

changeUserRole

```
virtual int changeUserRole(HRTCRoleType role, const char *authorization, long long ctime)
```

【功能说明】

设置用户在当前房间内的角色类型，角色切换时使用。

【请求参数】

- role：用户角色类型，joiner类型和player类型，具体请参见**HRTCRoleType**。
- authorization：预留参数，填null或者空字符串。
- ctime：预留参数，填0。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 加入房间前，可以通过**joinRoom**的joinParam参数确认角色信息。
- 加入指定房间后才可以在指定房间内进行角色切换，当前仅支持joiner和player角色切换。跨房场景下，通过对应connection连接下的changeUserRole接口实现在跨入房间中的角色类型切换。
- 切换成功触发**onUserRoleChangedNotify**回调。切换失败会触发**onError**回调，返回HRTC_ERR_CODE_USER_ROLE_CHANGE_FAIL错误码。
- 同一时间不同房间最多只能有一个joiner，player切换joiner的时候，需要将其他房间的joiner先切换成player。
- 不支持缺省用户昵称入会。

changeUserName

```
virtual int changeUserName(const char* userName)
```

【功能说明】

房间内设置用户自己的昵称。

【请求参数】

userName：变更的昵称。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 该接口仅支持房间内调用，更改的昵称会被实时同步到房间内其他用户的用户列表，退出房间不会保存，再次加入房间变更为加入房间时设置的昵称（参考**joinRoom**接口注意事项）。
- 会触发用户名变更的通知回调**onUserNameChangedNotify**。

createConnection

```
virtual IHRTCConnection* createConnection(const char* roomId, IHRTCConnectionEventHandler*
eventHandler) = 0)
```

【功能说明】

通过此接口可以创建一个与房间关联的IHRTCConnection连接对象，可多次调用以创建多个IHRTCConnection连接对象。调用每个连接对象中的joinRoom方法等接口，可以实现同时加入到多个房间，完成订阅和选看多个连接房间中的用户等功能。具体请参见**IHRTCConnection**和**事件回调（IHRTCConnectionEventHandler）**目录下的有关接口和回调。

【请求参数】

- roomId：房间ID。

- eventHandler：IHRTCConnectionEventHandler，引擎回调句柄，指定一个回调事件。SDK通过指定的事件通知应用程序的运行事件，如加入或离开房间等。具体请参见**5.3.4-事件回调（IHRTCConnectionEventHandler）**。

【返回参数】

IHRTCConnection：成功返回连接对象指针，失败返回内容为空。

⚠ 注意

同一时间最多只能创建4个连接对象，每个连接对象对应的房间ID必须互不相同。加上IHRTCEngine对象，即同时最多可加入5个房间，且在这5个房间中最多只能同时有一个joiner角色，其他只能为player角色。如果使用IHRTCEngine对象加入房间，则加入房间的房间ID不能和已创建连接对象对应的房间ID相同。

addMultiRoomMediaRelay

```
virtual int addMultiRoomMediaRelay(HRTCMultiRoomMediaRelayConfiguration
roomMediaRelayConfiguration)
```

【功能说明】

添加单个跨房。发起跨房后由云侧单向将本端上行流推至目标房间，即只推流不收流。

【请求参数】

roomMediaRelayConfiguration：跨房信息，具体请参见**HRTCMultiRoomMediaRelayConfiguration**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

removeMultiRoomMediaRelay

```
virtual int removeMultiRoomMediaRelay(HRTCMultiRoomMediaRelayConfiguration
roomMediaRelayConfiguration)
```

【功能说明】

删除单个跨房。

【请求参数】

roomMediaRelayConfiguration：跨房信息，具体请参见**HRTCMultiRoomMediaRelayConfiguration**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

stopMultiRoomMediaRelay

```
virtual int stopMultiRoomMediaRelay()
```

【功能说明】

停止所有跨房。

【请求参数】

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

7.4.1.5 音频管理

enableLocalAudioStream

```
virtual int enableLocalAudioStream(bool enabled)
```

【功能说明】

设置是否开启本地麦克风音频采集。

【请求参数】

enable：true表示开启。false表示关闭。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 默认开启本地音频采集。
- 远端用户会收到**onRemoteAudioStateChangedNotify**远端流状态变化回调。

adjustRecordingVolume

```
virtual int adjustRecordingVolume(unsigned int volume)
```

【功能说明】

设置麦克风采集的音量。

【请求参数】

volume：范围为[0-100]，其中10表示原始音量。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

不影响系统音量。

adjustPlaybackVolume

```
virtual int adjustPlaybackVolume(unsigned int volume)
virtual int adjustPlaybackVolume(const char* userId, unsigned int volume)
```

【功能说明】

设置扬声器播放的音量。

【请求参数】

volume：范围为[0-100]，其中10表示原始音量。

userId：用户ID。带用户ID表示设置单个用户的软（信号）音量。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

不影响系统音量。

setShareComputerSound

```
virtual int setShareComputerSound(bool enable)
```

【功能说明】

设置是否开启系统音频采集、发送。只能在房间内使用。

【请求参数】

enable：true表示开启系统音频采集、发送。false表示关闭系统音频采集、发送。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

开启系统音频采集发送，会采集和发送当前系统所有音频。

muteLocalAudio

```
virtual int muteLocalAudio(bool mute)
```

【功能说明】

设置是否关闭本地音频流发送。

【请求参数】

mute：true表示关闭本地音频发流。false表示开启本地音频发流。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 默认开启本地音频发流。
- 关闭本地音频发流，不影响本地音频采集。
- 远端用户会收到**onRemoteAudioStateChangedNotify**远端流状态变化回调。

muteRemoteAudio

```
virtual int muteRemoteAudio(const char* userId, bool mute)
```

【功能说明】

设置是否接收对应远端用户的音频流。同一时间所有房间最多只能接收17路音频流。

【请求参数】

- userId: 远端用户的userId，唯一标识。
- mute：true表示取消音频流接收。false表示开启音频流接收。默认为false。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 一般不需要接收自己的音频流，否则本端听筒里会听到自己说话的声音。
- TopN模式下，不支持此接口。

muteAllRemoteAudio

```
virtual int muteAllRemoteAudio(bool mute)
```

【功能说明】

设置是否接收当前房间所有用户的音频流。

【请求参数】

mute：true表示取消接收，false表示开启接收。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 取消所有音频流接收，同时也会取消接收新加入用户的音频流。
- 开启所有音频流接收，同时也会开启接收新加入用户的音频流。
- 默认开启所有音频流接收。
- 不支持TopN模式。

enableUserVolumeNotify

```
virtual int enableUserVolumeNotify(unsigned int interval)
```

【功能说明】

指定音量提示的时间间隔， 设置后按时间间隔触发回调，用户音量回调**onUserVolumeStatsNotify**。

【请求参数】

- interval：音量值上报周期，默认关闭音量回调。
  - 0：关闭音量回调。
  - [100，10000]：有效值范围，单位为毫秒。默认值为2000ms，建议设置为默认值2000ms。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setSpeakerModel

```
virtual int setSpeakerModel(HRTCSpeakerModel speakerModel)
```

【功能说明】

设置声音播放模式。成功加入房间后调用。

【请求参数】

speakerModel：声音播放模式（耳机/扬声器），具体请参见**HRTCSpeakerModel**。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HRTCErrorCode**。

enableAudioTNR

```
virtual int enableAudioTNR(bool enabled)
```

【功能说明】

设置是否开启音频降噪。

【请求参数】

enabled：true表示开启，false表示关闭。默认为false。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

sendAudioSeiMsg

```
virtual int sendAudioSeiMsg(const char* message, unsigned int msglen, unsigned int repeateCount)
```

【功能说明】

发送音频SEI消息。通过音频SEI可将自定义信息嵌入到音频流中，发送给其他用户。

【请求参数】

message：发送的内容。长度为1-500字节。

msglen : 发送内容的字节数。

repeateCount：发送次数（1-10）。根据需要填发送次数，一般发1次。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

echoTest

```
virtual int echoTest(int intervalInSeconds)
```

【功能说明】

开始语音通话回路测试。房间外调用，要等测试结束后才能加入房间。

【请求参数】

intervalInSeconds：当前只能设置为-1。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

finishEchoTest

```
virtual int finishEchoTest()
```

【功能说明】

停止语音通话回路测试。房间外调用。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setDefaultSpeakerModel

```
public abstract int setDefaultSpeakerModel(HRTCSpeakerModel speakerModel)
```

【功能说明】

设置默认的声音播放模式，在房间外调用。

【请求参数】

speakerModel：声音播放模式，具体请参见**HRTCSpeakerModel**。默认值为HRTC_SPEAKER_MODEL_SPEAKER。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

getAudioClipsVolume

```
virtual int getAudioClipsVolume()
```

【功能说明】

获取音效总音量。

【请求参数】

无

【返回参数】

- >=0：音量大小,范围0-100。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

音效文件支持同时播放多个，**setAudioClipsVolume**接口设置的音量是所有音频文件的最大音量，**setVolumeOfAudioClip**接口设置的是单个音效文件的音量，音效文件的实际播放音量 = 最大音量 * 自身音量 / 100。例如，最大音量是50，单个音效音量是80，实际播放音量就是50 * 80 / 100 = 40

setAudioClipsVolume

```
virtual int setAudioClipsVolume(double volume)
```

【功能说明】

设置音效总音量。

【请求参数】

volume：音量值大小，范围为0-100。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

getVolumeOfAudioClip

```
virtual int getVolumeOfAudioClip(int soundId)
```

【功能说明】

获取指定音效的播放音量。

【请求参数】

soundId：音效ID，取值>=0。

【返回参数】

- >=0：音量大小，范围为0-100。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

音效文件支持同时播放多个，**setAudioClipsVolume**接口设置的音量是所有音频文件的最大音量，**setVolumeOfAudioClip**接口设置的是单个音效文件的音量，音效文件的实际播放音量 = 最大音量 * 自身音量 / 100。例如，最大音量是50，单个音效音量是80，实际播放音量就是50 * 80 / 100 = 40

setVolumeOfAudioClip

```
virtual int setVolumeOfAudioClip(int soundId, double volume)
```

【功能说明】

设置指定音效音量。

【请求参数】

- soundId：音效ID，取值>=0。
- volume：音量大小，范围为0-100。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

playAudioClip

```
virtual int playAudioClip(int soundId, const char *filePath,int loop, double pitch, double pan, double gain, int publish, int startPos)
```

【功能说明】

播放音效文件并启动混音，需要在有joiner加入房间后调用。

【请求参数】

- soundId：音效ID，取值>=0。
- filePath：音效文件路径，支持本地文件和网络文件。
- loop：音效文件播放次数，0为不播放，-1为循环播放。
- pitch：音调大小，当前不支持。
- pan：空间位置，当前不支持。
- gain：音量大小，取值范围为0-100。
- publish：1表示将音效文件混音后发送到远端，0表示本地播放，不发送到远端。
- startPos：起始播放位置，单位为ms。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

soundId需要开发者自己生成并维护，保证不同的soundId对应不同的音效播放实例。同时音效播放完毕或者停止播放后，soundId最好主动回收，下一次播放音效的时候，尽量复用被回收的soundId。

stopAudioClip

```
virtual int stopAudioClip(int soundId)
```

【功能说明】

停止播放指定的音效文件。

【请求参数】

soundId：音效ID，取值>=0。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

stopAllAudioClips

```
virtual int stopAllAudioClips()
```

【功能说明】

停止播放所有音效文件。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

pauseAudioClip

```
virtual int pauseAudioClip(int soundId)
```

【功能说明】

暂停播放指定的音效文件。

【请求参数】

soundId：音效ID，取值>=0。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

pauseAllAudioClips

```
virtual int pauseAllAudioClips()
```

【功能说明】

暂停播放所有音效文件。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

resumeAudioClip

```
virtual int resumeAudioClip(int soundId)
```

【功能说明】

暂停播放指定的音效文件。

【请求参数】

soundId：音效ID，取值>=0。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

resumeAllAudioClips

```
virtual int resumeAllAudioClips()
```

【功能说明】

恢复播放所有音效文件。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

getAudioClipCurrentPosition

```
virtual int getAudioClipCurrentPosition(int soundId)
```

【功能说明】

获取指定音效文件当前的播放位置。

【请求参数】

soundId：音效ID，取值>=0。

【返回参数】

- >= 0：播放位置，单位为ms。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

setAudioClipPosition

```
virtual int setAudioClipPosition(int soundId, int pos)
```

【功能说明】

设置指定音效文件的播放位置。

【请求参数】

- soundId：音效ID，取值>=0。
- pos：播放位置，单位为ms。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

getAudioClipDuration

```
virtual int getAudioClipDuration(const char *filePath)
```

【功能说明】

获取音效的文件时长。

【请求参数】

filePath：音效文件路径，支持本地文件和网络文件。

【返回参数】

- > 0：音效文件时长，单位为ms。
- <= 0：方法调用失败。具体请参见**HRTCErrorCode**。

setAudioFrameRecordParameters

```
public abstract int setAudioFrameRecordParameters(int sampleRate, int channel, HRTCAudioOperateMode mode, int samplesPerCall);
```

【功能说明】

设置采集回调参数，配合setAudioFrameObserver的onAudioFrameRecord使用。

【请求参数】

- sampleRate：onAudioFrameRecord中返回的采样率，可设置为8000，16000，32000, 44100， 48000。
- channel：声道，1表示单声道，2表示双声道。
- mode：可读可写模式，具体请参见**HRTCAudioOperateMode**。

- samplesPerCall：每次回调的单声道样点数（小于(sampleRate/100)*channel*2*3, 大于(sample/(100*3))*channel*2）。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

7.4.1.6 视频管理

enableLocalVideo

```
virtual int enableLocalVideo(bool enabled)
```

【功能说明】

设置是否开启摄像头采集视频。

【请求参数】

enable：true表示开启。false表示关闭。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 默认开启本地视频采集。
- 远端用户订阅了本端用户时，会收到**onRemoteVideoStateChangedNotify**远端流状态变化回调。

setVideoEncoderConfig

```
virtual int setVideoEncoderConfig(const HRTCVideoEncParam &encParam)
virtual int setVideoEncoderConfig(unsigned int totalBitRate, HRTCVideoEncParam *encoderParams,
unsigned int encoderCount)
```

【功能说明】

接口一：大小流模式设置大流发流编码参数。大小流模式大流必须开启，小流建议开启。

接口二：多流模式设置发流编码参数。可支持720P到90P的四路流同时推送。

【请求参数】

encParam：视频编码参数。包括流类型、宽、高、码率、帧率等。其中宽必须是16的倍数，具体请参见**HRTCVideoEncParam**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 华为SDK系统有默认的编码设置（720P+360P），一般可以不设置发流编码参数。如果您确实需要自定义发流编码参数，请参考华为SDK系统推荐码表：**表7 不同分辨率的码率和帧率配置推荐**和**不同场景下帧率和码率的推荐值**，否则可能设置失败。
- 两重构接口针对不同使用方式并不通用，如果您使用的新选看系列接口**setupRemoteView**，需配套使用接口一：大小流模式设置发流编码参数的**setVideoEncoderConfig**，**enableSmallVideoStream**接口来自定义您的发流编码参数。同理使用旧的选看接口**startRemoteStreamView**需配套使用接口二：多流模式设置发流编码参数setVideoEncoderConfig接口。
- 同一端大小流或多流模式多路流设置的分辨率需保持一致，否则会设置失败。
- 多终端发流和选看的分辨率不一致时，sdk默认自适应并匹配最接近的分辨率（以实际发流分辨率优先），可能会导致选看时设置的分辨率和实际收到的流分辨率不一致。
- 调用接口一设置编码参数的分辨率发生变化时，需要先**enableSmallVideoStream**关闭小流，否则会因为分辨率一致条件限制导致大流设置失败。即涉及分辨率变化时，先关闭小流，再设置大流，再设置小流。

enableSmallVideoStream

```
virtual int enableSmallVideoStream(bool enable, const HRTCVideoEncParam &encParam)
```

【功能说明】

大小流模式设置是否开启小流，并设置编码参数。小流可关闭但一般建议开启。

【请求参数】

- enable：是否开启小流。
- encParam：视频编码参数。包括流类型、宽、高、码率、帧率等。具体请参见**HRTCVideoEncParam**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

startLocalPreview

```
virtual int startLocalPreview();
```

【功能说明】

开始本地预览。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 在房间外预览：先调用startLocalPreview开启预览，成功后再调用**setupLocalView**设置有效view实现预览；进入房间前需调用stopLocalPreview关闭预览，否则将一直处于预览状态，影响房间内预览。
- 在房间内预览：可调用**setupLocalView**设置有效view开启预览，设置为0表示关闭预览，不需要调用startLocalPreview和stopLocalPreview。

stopLocalPreview

```
virtual int stopLocalPreview();
```

【功能说明】

停止本地预览。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

如果调用了**startLocalPreview**，需调用stopLocalPreview关闭预览，否则将一直处于预览状态。该接口限制在房间外调用，在房间内设置不生效。

setupLocalView

```
virtual int setupLocalView(view_t view, HRTCVideoDisplayMode displayMode)
virtual int setupLocalView(view_t view)
```

【功能说明】

设置本地渲染视图。该方法设置本地视频显示信息。 App通过调用此接口绑定本地视频流的显示视窗(view)，并设置视频显示模式。

【请求参数】

- view：窗口句柄。
- displayMode：图像填充模式。具体请参见**HRTCVideoDisplayMode**。不设置则默认为裁剪模式。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

updateLocalRenderMode

```
virtual int updateLocalRenderMode(HRTCVideoDisplayMode displayMode, HRTCVideoMirrorType
mirrorMode) = 0;
```

【功能说明】

设置本地视图渲染模式，镜像模式。

【请求参数】

- displayMode：渲染模式。具体请参见**HRTCVideoDisplayMode**。
- mirrorMode：镜像模式。具体请参见**HRTCVideoMirrorType** 。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

> **⚠ 注意**
>
> 当前Windows平台的displayMode参数是无效的，只能使用**setupLocalView**设置本地渲染模式。

setupRemoteView

```
virtual int setupRemoteView(const char* userId, view_t view)
```

【功能说明】

设置远端流渲染视图（新选看接口），该接口不影响收流。

【请求参数】

- userId：远端用户的唯一标识。
- view：窗口句柄，view为NULL时，解除窗口绑定并结束选看。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

updateRemoteRenderMode

```
virtual int updateRemoteRenderMode(const char* userId, HRTCVideoDisplayMode displayMode,
HRTCVideoMirrorType mirrorMode)
```

【功能说明】

设置远端用户视图渲染模式。

【请求参数】

- userId：远端用户的唯一标识。
- displayMode：视图显示模式。具体请参见**HRTCVideoDisplayMode**，默认RTC_VIDEO_DISPLAY_HIDDEN，通过裁剪的方式保持宽高比。
- mirrorMode：镜像模式。具体请参见**HRTCVideoMirrorType**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setRemoteVideoStreamType

```
virtual int setRemoteVideoStreamType(const char *userId, HRTCVideoStreamType type)
```

【功能说明】

大小流模式，设置指定选看用户的视频流类型。在通过新选看接口发起选看时调用。

【请求参数】

- userId：远端用户唯一标识。
- type：视频流类型。指大流、小流，具体请参见**HRTCVideoStreamType**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setPriorRemoteVideoStreamType

```
virtual int setPriorRemoteVideoStreamType(HRTCVideoStreamType type)
```

【功能说明】

大小流模式，设置所有订阅的远端视频流类型。默认订阅大流，优先使用**setRemoteVideoStreamType**接口设置的用户流类型。

【请求参数】

type：视频流类型。指大流、小流，具体请参见**HRTCVideoStreamType**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

pushLocalVideo

```
virtual int pushLocalVideo(bool push)
```

【功能说明】

设置是否开启发送本地视频流。

【请求参数】

push：true表示开启，false表示关闭。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 默认开启本地视频发流。
- 关闭本地视频发流，不影响本地视图采集，仍可见本地视图。
- 远端用户订阅了本端用户时，会收到**onRemoteVideoStateChangedNotify**远端流状态变化回调。

pullRemoteVideo

```
virtual int pullRemoteVideo(const char* userId, bool pull)
```

【功能说明】

开启/关闭接收指定远端用户的视频流。只能加入房间后调用。

【请求参数】

- userId：远端用户的userId，唯一标识。
- pull：true表示开始接收，false表示关闭接收。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

pullAllRemoteVideo

```
virtual int pullAllRemoteVideo(bool pull)
```

【功能说明】

开启/关闭接收当前房间所有远端用户的视频流。

【请求参数】

pull：true表示开始接收，false表示关闭接收， 默认开启接收。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

startRemoteStreamView

```
virtual int startRemoteStreamView(const char* userId, view_t view, HRTCStreamType streamType, bool disableAdjustRes)
```

【功能说明】

设置远端用户渲染视图，并开启收流（选看）。

【请求参数】

- userId：远端用户的唯一标识。
- view：窗口句柄。
- streamType：视频分辨率，具体请参见**HRTCStreamType**。

- disableAdjustRes：禁用分辨率自适应，默认值为false，开启分辨率自适应。若关闭，在网络环境较差情况下可能会有卡顿现象。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

注意

该接口为选看的旧接口，通过该接口和updateRemoteRenderMode完成一次完整的选看流程。新的完成选看功能拆分为三个接口：**setupRemoteView**、**pullRemoteVideo**和**setRemoteVideoStreamType**接口，将设置渲染模式、窗口句柄、选看的流类型拆分并增加pullRemoteVideo收流控制接口，以实现更细化的选看流程控制（将窗口绑定和收流控制分开）。您可以根据需要选择调用不同的接口组合以实现视频选看。

stopRemoteStreamView

```
virtual int stopRemoteStreamView(const char* userId)
```

【功能说明】

关闭远端用户的渲染视图，并停止收流（停止选看）。

【请求参数】

userId：远端用户的唯一标识。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

注意

新选看建议通过**setupRemoteView**接口设置窗口句柄为空来停止选看。

setRemoteVideoAdjustResolution

```
virtual int setRemoteVideoAdjustResolution(bool enable)
```

【功能说明】

设置是否开启远端流分辨率自适应。默认开启自适应。

【请求参数】

enable：是否开启自适应。默认开启。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

只改变本地摄像头预览视图，不改变远端观看视图。预览时设置不生效。

setVideoEncoderMirror

```
virtual int setVideoEncoderMirror(HRTCVideoMirrorType mirrorType)
```

【功能说明】

设置编码器输出的（本地发流）画面镜像模式。

【请求参数】

mirrorType：镜像模式，是否开启镜像。具体请参见**HRTCVideoMirrorType**。默认值为HRTC_VIDEO_MIRROR_TYPE_DISABLE。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

该接口不改变本地摄像头的预览画面，但会使远端用户看到的和服务器录制的画面为指定的镜像效果。

setCameraConfig

```
virtual int setCameraConfig(HRTCCameraConfig &config)
```

【功能说明】

设置相机参数。

【请求参数】

config：摄像头参数，参考**HRTCCameraConfig**。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HRTCErrorCode**。

switchCamera

```
virtual int switchCamera()
```

【功能说明】

切换摄像头，开启摄像头后生效。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HRTCErrorCode**。

startAllRemoteView

```
virtual int startAllRemoteView(unsigned int counts, const HRTCVideoRemoteView *viewInfo)
```

【功能说明】

批量设置远端流视图。

【请求参数】

- counts：订阅的视图数量，如果设置为0，则取消所有远端流视图。
- viewInfo：订阅的视图信息，主要包括代表该视图的句柄、流类型、用户ID、是否自适应等，具体请参见**HRTCVideoRemoteView**。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HRTCErrorCode**。

setRemoteViewRotation

```
virtual int setRemoteViewRotation(const char* userId, HRTCVideoRotation rotation)
```

【功能说明】

设置远端流视图的旋转角度。

【请求参数】

- userId：用户ID。
- rotation：旋转角度信息（0°，90°，270°），具体请参见**HRTCVideoRotation**。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HRTCErrorCode**。

setRemoteViewOrientation

```
virtual int setRemoteViewOrientation(const char *userId, HRTCVideoOrientation orientation)
```

【功能说明】

设置远端流视图方向（横竖屏）。

【请求参数】

- userId：用户ID。
- orientation：方向（横竖屏），具体请参见**HRTCVideoOrientation**。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HRTCErrorCode**。

setVideoPaddingImage

```
virtual int setVideoPaddingImage(const char* imagePath, int fps)
```

【功能说明】

设置关闭视频发流时发送的图片。

【请求参数】

- imagePath：图片路径，当前限制为bmp格式图片。
- fps：发送的帧率，范围为5-20帧，建议值为10帧。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HRTCErrorCode**。

setBeautyRetouchOption

```
virtual int  setBeautyRetouchOption(boolean enabled, HRTCBeautyOptions option)
```

【功能说明】

设置是否开启美颜功能。

【请求参数】

- enabled：true表示打开，false表示关闭。
- option : 美颜设置参数，具体请参见**HRTCBeautyOptions**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

enableVideoBrighten

```
virtual int enableVideoBrighten(bool enabled)
```

【功能说明】

设置是否开启视频增亮功能。

【请求参数】

enabled：true表示开启，false表示关闭。

ApplyGSensorMode

```
virtual int ApplyGSensorMode(HRTCGSensorMode mode)
```

【功能说明】

是否开启重力感应。

【请求参数】

HRTCGSensorMode：重力感应模式，具体请参见**HWRTCGSensorMode**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

startPublishStream

```
virtual int startPublishStream(const char* taskId, const HRTCRtmpUrlList &urlList, const HRTCTranscodeConfig &transcodeConfig)
```

【功能说明】

开始旁路推流。

【请求参数】

- taskId：任务id，用户自定义，需保证唯一。
- urlList：url数组，具体请参见**HRTCRtmpUrlList**。
- transcodeConfig：用户id数组和其他参数，具体请参见**HRTCTranscodeConfig**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

updateTransCoding

```
virtual int updateTransCoding(const char* taskId,  const HRTCTranscodeConfig &transcodeConfig)
```

【功能说明】

更新旁路推流。收到远端用户重新入会时，需要重新推流。

【请求参数】

- taskId：任务id，用户自定义，需保证唯一。
- transcodeConfig：用户id数组和其他参数，具体请参见**HRTCTranscodeConfig**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

stopPublishStream

```
virtual int stopPublishStream(const char* taskId)
```

【功能说明】

停止旁路推流。

【请求参数】

taskId：任务id，用户自定义，需保证唯一。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setVideoWaterMark

```
virtual int setVideoWaterMark(const char *imagePath, float xOffset, float yOffset, float fWidthRatio)
```

【功能说明】

插入或删除水印，本接口支持为主流和辅流添加水印。

【请求参数】

- imagePath：水印图片本地路径（传nullptr表示去掉水印）。
- xOffset：水印显示的左上角X轴偏移，取(0,1)的浮点数。
- yOffset：水印显示的左上角y轴偏移，取(0,1)的浮点数。
- fWidthRatio：水印显示的宽度占画面宽度的比例（水印按该参数等比例缩放显示），取(0,1)的浮点数。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setBackgroundBlur

```
virtual int setBackgroundBlur(bool enable, int radius) = 0;
```

【功能说明】

设置本地视频背景虚化功能，与背景替换功能互斥。

【请求参数】

- enable：开关。
- radius：虚化半径，范围为0-25。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setBackgroundReplace

```
virtual int setBackgroundReplace(bool enable, const char* imagePath) = 0;
```

【功能说明】

设置本地视频背景替换功能，与背景虚化功能互斥。

【请求参数】

- enable：开关。
- imagePath：背景替换的图片路径，有效路径长度为255（包含结束符），支持格式为jpg、jpeg、png。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

appendLocalView

```
virtual int appendLocalView(view_t view) = 0;
```

【功能说明】

设置本地视频另一个窗口显示。只有在SDK渲染的模式下有效。

【请求参数】

view：窗口句柄。 null表示取消扩展的窗口显示。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

appendRemoteView

```
virtual int appendRemoteView(const char *userId, view_t view) = 0;
```

【功能说明】

设置远程视频另一个窗口显示。只有在SDK渲染的模式下有效。

【请求参数】

- view：窗口句柄。 null表示取消扩展的窗口显示。
- userid：用户ID。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

7.4.1.7 辅流管理

setRemoteAuxiliaryStreamViewRotation

```
virtual int setRemoteAuxiliaryStreamViewRotation(const char *userId, HRTCVideoRotation rotation)
```

【功能说明】

设置远端辅流视图旋转角度。

【请求参数】

- userId：用户ID。
- rotation：旋转角度信息（0°，90°，270°），具体请参见**HRTCVideoRotation**。默认为0°。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HRTCErrorCode**。

setRemoteAuxiliaryStreamViewOrientation

```
virtual int setRemoteAuxiliaryStreamViewOrientation(const char *userId, HRTCVideoOrientation orientation)
```

【功能说明】

设置远端辅流视图方向（横竖屏）。

【请求参数】

- userId：用户ID。
- orientation：方向（横竖屏），具体请参见**HRTCVideoOrientation**。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HRTCErrorCode**。

startRemoteAuxiliaryStreamView

```
virtual int startRemoteAuxiliaryStreamView(const char* userId, view_t view)
```

【功能说明】

当远端开启辅流，本地接收到远端辅流开启**onUserAuxiliaryStreamAvailable**消息后，设置辅流流窗口视图（发起辅流选看）。

【请求参数】

- userId：远端用户的唯一标识。
- view ： 窗口句柄。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 收到onUserAuxiliaryStreamAvailable消息后，获取对应的userId。
- 多辅流场景，一个用户同时只能订阅一条辅流；当前正在订阅用户A的辅流，需要订阅另一个用户B的辅流时，需要先停止订阅用户A的辅流，再订阅用户B的辅流。

stopRemoteAuxiliaryStreamView

```
virtual int stopRemoteAuxiliaryStreamView(const char* userId)
```

【功能说明】

关闭屏幕辅流窗口视图（停止辅流选看）。

【请求参数】

userId：远端用户的唯一标识。对应**onUserAuxiliaryStreamAvailable**返回的共享用户标识。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

收到**onUserAuxiliaryStreamAvailable**消息后，如果选看的远端辅流不可用，则必须调用stopRemoteAuxiliaryStreamView关闭。

updateRemoteAuxiliaryStreamRenderMode

```
virtual int updateRemoteAuxiliaryStreamRenderMode(const char* userId, HRTCVideoDisplayMode displayMode, HRTCVideoMirrorType mirrorMode)
```

【功能说明】

设置辅流视图渲染模式。

【请求参数】

- userId：远端用户的唯一标识。
- displayMode：视图显示模式 。具体请参见**HRTCVideoDisplayMode**，默认RTC_VIDEO_DISPLAY_FIT，通过扩边的方式保持宽高比。
- mirrorMode：镜像模式。具体请参见**HRTCVideoMirrorType**。默认为HRTC_VIDEO_MIRROR_TYPE_DISABLE。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setAuxiliaryVideoEncodeSmooth

```
virtual int setAuxiliaryVideoEncodeSmooth(bool enable)
```

【功能说明】

设置是否开启辅流的流畅度优先（降低辅流选看分辨率）。

【请求参数】

enable：true表示辅流分辨率为720p，false表示辅流分辨率为1080p。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

开启后，辅流发流分辨率为720p，否则发流分辨率为1080p。默认不开启。

setAuxiliaryVideoEncoderConfig

```
virtual int setAuxiliaryVideoEncoderConfig(HRTCVideoAuxiliaryEncParam &encoderParams)
```

【功能说明】

设置辅流编码参数。

【请求参数】

encoderParams：需要设置的辅流编码参数，包括宽、高、帧率、码率，具体请参见HRTCVideoAuxiliaryEncParam。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见HRTCErrorCode。

⚠ 注意

使用该接口设置辅流编码参数，目前策略会根据获取的区域宽高比对设置的宽高进行调整，使得用户设置的宽高比与获取宽高比保持一致，这样用户实际收流分辨率与设置分辨率有可能不同。

7.4.1.8 屏幕共享

getScreenShareSources

```
virtual IHRTCScreenShareSourceList* getScreenShareSources(HRTCScreenShareIconType type)
```

【功能说明】

获取屏幕可共享对象列表。

【请求参数】

type：屏幕捕获图像类型，具体请参见HRTCScreenShareIconType。

【返回参数】

共享屏幕窗口对象列表，具体请参见共享屏幕资源管理。

setScreenShareTarget

```
virtual int setScreenShareTarget(HRTCScreenShareSourceInfo* info, HRTCSrceenCaptureOptionalInfo* optionalInfo)
```

【功能说明】

选择屏幕共享对象。

【请求参数】

- info：共享对象信息，主要为采集源ID、名称等，具体请参见HRTCScreenShareSourceInfo。
- optionalInfo：其他共享对象信息，如是否禁止鼠标采集、可选的共享区域等，具体请参见HRTCSrceenCaptureOptionalInfo。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见HRTCErrorCode。

⚠ 注意

- 辅流的分辨率默认最大支持1080P，如果小于1080P，以实际分辨率为准。
- 区域共享分辨率以optionalInfo中定义的矩形（Rect）大小为准，若不设置（全部设为0），则默认使用1920*1080。
- 区域共享时自定义分辨率的矩形（Rect）范围限制为最小96*92，最大4096*2160，超出范围设置无效。

startScreenShare

```
virtual int startScreenShare()
```

【功能说明】

开启屏幕共享。开启屏幕共享前需要调用**getScreenShareSources**和**setScreenShareTarget**接口获取和选择共享屏幕（窗口）对象。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 目前可支持多路辅流共享，若需开启多辅流，请**提交工单**联系技术支持处理。
- 共享成功后会触发onScreenShareStarted回调。
- 远端会收到**onUserAuxiliaryStreamAvailable**通知，可据此发起辅流选看。

stopScreenShare

```
virtual int stopScreenShare()
```

【功能说明】

停止屏幕共享。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 停止成功后会触发**onScreenShareStopped**回调。
- 远端会收到**onUserAuxiliaryStreamAvailable**通知，可据此停止辅流选看。

addHiddenShareWindow

```
virtual int addHiddenShareWindow(view_t view)
```

【功能说明】

将指定窗口加入屏幕共享排除列表，屏幕共享时，在排除列表中的窗口将不会被共享出去。

屏幕共享前后均可调用。

【请求参数】

view：窗口句柄。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 添加过滤窗口，窗口关闭后，不会自动移除该窗口句柄。
- 该功能仅在屏幕共享时生效，在窗口共享时不生效。
- 退出房间不会自动清空过滤窗口列表。
- windows7系统，通过DwmIsCompositionEnabled查询为关闭的场景下不可用。
- windows 10系统1607以前的版本，通过GetProcessDpiAwareness查询为PROCESS_DPI_UNAWARE或者PROCESS_SYSTEM_DPI_AWARE，则在系统dpi非100%场景不可用。建议应用以PROCESS_PER_MONITOR_DPI_AWARE模式使用。

deleteHiddenShareWindow

```
virtual int deleteHiddenShareWindow(view_t view)
```

【功能说明】

将指定窗口从屏幕共享排除列表中移除。

屏幕共享前后均可调用。

【请求参数】

view：窗口句柄。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

removeAllHiddenShareWindow

```
virtual int removeAllHiddenShareWindow()
```

【功能说明】

将所有窗口从屏幕共享排除列表中移除。

屏幕共享前后均可移除。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

getScreenShareSources

```
virtual IHRTCScreenShareSourceList*  getScreenShareSources(const HRTCSize &thumbnailSize, const HRTCSize &iconSize)
```

【功能说明】

获取屏幕可共享对象列表，包含缩略图。

【请求参数】

thumbnailSize：具体请参见**HRTCSize**，当前只支持640*360， 传0*0时不需要缩略图。

iconSize：具体请参见**HRTCSize**，当前不支持。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setVideoEncodeResolutionMode

```
virtual int setVideoEncodeResolutionMode(HRTCVideoEncodeResolutionMode resolutionMode)
```

【功能说明】

设置视频编码分辨率比例模式。

【请求参数】

resolutionMode：视频编码分辨率比例模式。具体请参见**HRTCVideoEncodeResolutionMode**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

使用该接口设置辅流编码参数，目前策略会根据获取的区域宽高比对设置的宽高进行调整，使用户设置的宽高比与获取的宽高比保持一致，因此，用户实际收流分辨率与设置分辨率有可能不同。

7.4.1.9 音频文件播放管理

音频文件播放使用ffmpeg能力，在播放在线url时受windows平台自身特性影响，断网时会立刻停止播放（SparkRTC内部无缓存设计），若对该场景体验有要求，建议业务通过先下载到本地再播放的形式使用。

startAudioFile

```
virtual int startAudioFile(const char *filePath, int publish, int cycle, int replace)
virtual int startAudioFile(const char *filePath, int publish, int cycle, int replace, unsigned int startPos)
```

【功能说明】

播放音频文件，房间内调用。当前仅支持本端播放。

【请求参数】

- filePath：音频文件的本地全路径。
- publish：播放模式，0表示只有本端能听到播放的音频，1表示远端也能听到播放的音频。
- cycle：循环次数，0表示无限循环。
- replace：远端模式下面是否需要和麦克风做混音。
- startPos：音频文件开始播放的位置，单位为ms。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

会触发**onAudioMixStateChangedNotify**回调。

stopAudioFile

```
virtual int stopAudioFile()
```

【功能说明】

停止播放音频文件，房间内调用。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

会触发**onAudioMixStateChangedNotify**回调。

pauseAudioFile

```
virtual int pauseAudioFile()
```

【功能说明】

暂停播放音频文件，房间内调用。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- <0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

会触发**onAudioMixStateChangedNotify**回调。

resumeAudioFile

```
virtual int resumeAudioFile()
```

【功能说明】

恢复播放音频文件，房间内调用。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- <0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

会触发**onAudioMixStateChangedNotify**回调。

adjustAudioFileVolume

```
int adjustAudioFileVolume(unsigned int volume)
```

【功能说明】

调节混音里伴奏在本端和远端播放的音量。

【请求参数】

volume：音量大小，范围为0-100。默认音量为100。

【返回参数】

- 0：方法调用成功。
- <0：方法调用失败。具体请参见**HRTCErrorCode**。

adjustAudioFilePlayoutVolume

```
int adjustAudioFilePlayoutVolume(unsigned int volume)
```

【功能说明】

调节混音里伴奏在本端播放的音量大小。

【请求参数】

volume：音量大小，范围为0-100。默认音量为100。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

getAudioFileVolume

```
int getAudioFileVolume()
```

【功能说明】

获取音频播放的音量。

【请求参数】

无

【返回参数】

- >=0：音量大小，音量范围为0-100。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

getAudioFilePlayoutVolume

```
int getAudioFilePlayoutVolume()
```

【功能说明】

获取本地音频播放的音量。

【请求参数】

无

【返回参数】

- >=0：音量大小，音量范围为0-100。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

getAudioFileDuration

```
int getAudioFileDuration()
```

【功能说明】

获取音频播放的时长。

【请求参数】

无

【返回参数】

- >0：音频时长，单位为ms。
- <= 0：方法调用失败。具体请参见**HRTCErrorCode**。

getAudioFilePosition

```
int getAudioFilePosition()
```

【功能说明】

获取音频文件当前播放位置。

【请求参数】

无

【返回参数】

- >=0：音频时长，单位为ms。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

setAudioFilePosition

```
int setAudioFilePosition(unsigned long long position)
```

【功能说明】

设置音频文件播放位置。

【请求参数】

position：播放位置，单位为ms。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

getAudioClipsVolume

```
virtual int getAudioClipsVolume()
```

【功能说明】

获取音效总音量。

【请求参数】

无

【返回参数】

- >=0：音量大小，范围为0-100。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

音效文件支持同时播放多个，**setAudioClipsVolume**接口设置的音量是所有音频文件的最大音量，**setVolumeOfAudioClip**接口设置的是单个音效文件的音量，音效文件的实际播放音量 = 最大音量 * 自身音量 / 100。例如，最大音量是50，单个音效音量是80，实际播放音量就是50 * 80 / 100 = 40

setAudioClipsVolume

```
virtual int setAudioClipsVolume(double volume)
```

【功能说明】

设置音效总音量。

【请求参数】

volume：音量值大小，范围为0-100。默认音量为100。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

getVolumeOfAudioClip

```
virtual int getVolumeOfAudioClip(int soundId)
```

【功能说明】

获取指定音效的播放音量。

【请求参数】

soundId：音效ID，取值>=0。

【返回参数】

- >=0：音量大小，范围为0-100。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

支持同时播放多个音效文件，**setAudioClipsVolume**接口设置的音量是所有音频文件的最大音量，**setVolumeOfAudioClip**接口设置的是单个音效文件的音量，音效文件的实际播放音量 = 最大音量 * 自身音量 / 100。例如，最大音量是50，单个音效音量是80，实际播放音量就是50 * 80 / 100 = 40

setVolumeOfAudioClip

```
virtual int setVolumeOfAudioClip(int soundId, double volume)
```

【功能说明】

设置指定音效音量。

【请求参数】

- soundId：音效ID，取值>=0。
- volume：音量大小，范围为0-100。默认音量为100。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

playAudioClip

```
virtual int playAudioClip(int soundId, const char *filePath,int loop, double pitch, double pan, double gain, int publish, int startPos)
```

【功能说明】

播放音效文件并启动混音，需要在有joiner加入房间后调用。

【请求参数】

- soundId：音效ID，取值>=0。
- filePath：音效文件路径，支持本地文件和网络文件。
- loop：音效文件播放次数，0为不播放，-1为循环播放。
- pitch：音调大小，当前不支持。
- pan：空间位置，当前不支持。
- gain：音量大小，取值范围为0-100。
- publish：1表示将音效文件混音后发送到远端，0为本地播放，不发送到远端。
- startPos：起始播放位置，单位为ms。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

soundId需要开发者自己生成并维护，保证不同的soundId对应不同的音效播放实例。同时音效播放完毕或者停止播放后，soundId最好主动回收，下一次播放音效的时候，尽量复用被回收的soundId。

stopAudioClip

```
virtual int stopAudioClip(int soundId)
```

【功能说明】

停止播放指定的音效文件。

【请求参数】

soundId：音效ID，取值>=0。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

stopAllAudioClips

```
virtual int stopAllAudioClips()
```

【功能说明】

停止播放所有音效文件。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

pauseAudioClip

```
virtual int pauseAudioClip(int soundId)
```

【功能说明】

暂停播放指定的音效文件。

【请求参数】

soundId：音效ID，取值>=0。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

pauseAllAudioClips

```
virtual int pauseAllAudioClips()
```

【功能说明】

暂停播放所有音效文件。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

resumeAudioClip

```
virtual int resumeAudioClip(int soundId)
```

【功能说明】

暂停播放指定的音效文件。

【请求参数】

soundId：音效ID，取值>=0。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

resumeAllAudioClips

```
virtual int resumeAllAudioClips()
```

【功能说明】

恢复播放所有音效文件。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

getAudioClipCurrentPosition

```
virtual int getAudioClipCurrentPosition(int soundId)
```

【功能说明】

获取指定音效文件当前的播放位置。

【请求参数】

soundId：音效ID，取值>=0。

【返回参数】

- >= 0：播放位置，单位为ms。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

setAudioClipPosition

```
virtual int setAudioClipPosition(int soundId, int pos)
```

【功能说明】

设置指定音效文件的播放位置。

【请求参数】

- soundId：音效ID，取值>=0。
- pos：播放位置，单位为ms。

【返回参数】

- 0：方法调用成功。
- < 0：方法调用失败。具体请参见**HRTCErrorCode**。

getAudioClipDuration

```
virtual int getAudioClipDuration(const char *filePath)
```

【功能说明】

获取音效的文件时长。

【请求参数】

filePath：音效文件路径，支持本地文件和网络文件。

【返回参数】

- > 0：音效文件时长，单位为ms。
- <= 0：方法调用失败。具体请参见**HRTCErrorCode**。

7.4.1.10 自采集自渲染

setExternalVideoCapture

```
virtual int setExternalVideoCapture(bool enable, HRTCVideoFrameFormat format)
```

【功能说明】

设置是否开启外部视频采集。

【请求参数】

- enable：true表示开启视频自采集，false表示取消视频自采集。默认值为false。
- format：设置外部采集的视频格式，默认为I420（即yuv420P）。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 加入房间前调用，不支持房间内切换。
- 视频支持格式：I420，texture2d。
- 开启视频自采集后不支持视频预览。

pushExternalVideoFrame

```
virtual int pushExternalVideoFrame(HRTCVideoFrame* videoFrame)
```

【功能说明】

推送外部视频数据。

【请求参数】

videoFrame：视频自采集数据格式。具体请参见**HRTCVideoFrame**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

注意

- 此方法调用前，需要先调用**setExternalVideoCapture**设置开启外部视频采集。
- 数据输入周期：同视频周期，1/帧率。

setExternalVideoFrameOutput

```
virtual int setExternalVideoFrameOutput(bool localEnable, bool remoteEnable, HRTCImageBufferFormat format = HRTCImageBufferFormat())
```

【功能说明】

设置是否开启视频流自渲染。开启后，回调**onRenderExternalVideoFrame**中会有视频帧数据上报。

【请求参数】

- localEnable：开启本地视频自渲染，默认sdk渲染。
- remoteEnable：开启远端视频自渲染，默认sdk渲染。
- format：默认imageFormat为HRTC_VIDEO_IMAGE_FORMAT_YUV420P，bufferType为HRTC_VIDEO_IMAGE_BUFFER_BYTE_ARRAY。具体请参见**HRTCImageBufferFormat**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

注意

自渲染模式下，仍需要调用**startRemoteStreamView**，接收到远端视频数据。

setAuxiliaryExternalVideoFrameOutput

```
virtual int setAuxiliaryExternalVideoFrameOutput(bool localEnable, bool remoteEnable)
```

【功能说明】

设置是否开启辅流自渲染。开启后，回调**onRenderAuxiliaryExternalVideoFrame**中会有辅流数据上报。

【请求参数】

- localEnable：开启本地辅流自渲染，默认sdk渲染。
- remoteEnable：开启远端辅流自渲染，默认sdk渲染。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

辅流暂不支持本地渲染，该方法只对远端辅流生效。

setExternalAudioCapture

```
virtual int setExternalAudioCapture(int enabled, int sampleRate, int channels)
```

【功能说明】

设置是否开启外部音频采集。

【请求参数】

- enable：true表示开启音频自采集，false表示取消音频自采集。
- sampleRate：音频采样率，支持16k/48k。
- channels：音频声道数，支持单声道。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 加入房间前调用，不支持房间内切换。
- 自采集音频输入规格：
  - 格式：PCM。
  - 采样率：16k/48k 。
  - 声道数：单声道。
  - 位数：16。

pushExternalAudioFrame

```
virtual int pushExternalAudioFrame(void* audioData, int size)
```

【功能说明】

推送外部音频数据。

【请求参数】

- audioData：音频数据。
- size：音频输入数据大小。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 此方法调用前，需要先调用**setExternalAudioCapture**设置开启外部音频采集。
- 数据输入周期：10ms。
- 音频输入数据大小：10 * sampleRate * channels * 16 / 8 / 1000。

setExternalAudioFrameOutput

```
virtual int setExternalAudioFrameOutput(bool localEnable, bool remoteEnable)
```

【功能说明】

设置是否开启音频自渲染。开启后，回调**onPlaybackExternalAudioFrame**中会有音频帧数据上报。

【请求参数】

- localEnable：开启本地音频自渲染，默认sdk渲染。
- remoteEnable：开启远端音频自渲染，默认sdk渲染

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

7.4.1.11 其他接口

startNetworkTest

```
virtual int startNetworkTest(const HRTCNetworkTestConfig* networkTestConfig) = 0;
```

【功能说明】

开启网络质量测试，房间外调用，要等探测结束后才能加入房间。

【请求参数】

networkTestConfig：网络探测参数格式。具体请参见**HRTCNetworkTestConfig**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

会触发**onNetworkTestQuality**和**onNetworkTestResult**两个回调，并通过回调返回具体的网络测试结果，探测大约需要20-60s。

stopNetworkTest

```
virtual int stopNetworkTest()
```

【功能说明】

停止网络质量测试，房间外调用。

【请求参数】

无

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setAudioConfig

```
virtual int setAudioConfig(HRTCAudioQualityLevel level, HRTCAudioSceneType scene)
```

【功能说明】

设置使用场景。该接口需要在**joinRoom**前调用。

此接口可在initialize接口设置场景后改变音频场景，暂不支持初始化scene设置音乐再通过此接口设置为会议。

【请求参数】

level：表示档位，会议模式暂时只支持16k。具体请参见**HRTCAudioQualityLevel**。

scene：表示音频场景模式，具体请参见**HRTCAudioSceneType**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

7.4.2 事件回调（IHRTCEngine）

本章节介绍了全平台C++ SDK的回调接口IHRTCEngineEventHandler的详情。

表 7-10 事件回调说明

接口	描述
onError	错误回调。
onJoinRoomSuccess	成功加入房间回调。
onJoinRoomFailure	加入房间失败回调。
onLeaveRoom	离开房间回调。
onRemoteUserOnline	远端用户加入当前房间回调。
onRemoteUserOffline	远端用户离开当前房间回调。
onRemoteUserNameChangedNotify	远端用户昵称变化回调。
onUserNameChangedNotify	本地用户昵称变化回调。
onFirstRemoteVideoDecoded	引擎收到第一帧远端视频流并解码成功回调。

接口	描述
onFirstRemoteAuxiliaryStreamDecoded	引擎收到第一帧远端辅流并解码成功回调。
onConnectionChangedNotify	网络连接状态发生变化回调。
onDeviceStateChangedNotify	设备状态发生变化回调。
onDeviceVolumeChangedNotify	音频设备音量发生变化回调。
onLogUploadResult	日志上传结果回调。
onLogUploadProgress	日志上传进度回调。
onUserRoleChangedNotify	用户角色切换成功回调。
onScreenShareStarted	辅流开启回调。
onScreenShareStopped	辅流关闭回调。
onUserAuxiliaryStreamAvailable	远端开启/停止辅流回调。
onVideoStatsNotify	视频流详情，2s触发一次回调。
onAudioStatsNotify	音频流详情，2s触发一次回调。
onAuxiliaryStreamStatsNotify	辅流详情，2s触发一次回调。
onAuthorizationExpired	鉴权签名过期回调。
onRemoteAudioStateChangedNotify	远端音频流状态变化回调。
onRemoteVideoStateChangedNotify	远端视频流状态变化回调。
onRenderAuxiliaryExternalVideoFrame	辅流自渲染回调。
onRenderExternalVideoFrame	视频自渲染回调
onPlaybackExternalAudioFrame	音频渲染回调
onRejoinRoomSuccess	重新加入房间回调。
onNetworkTestQuality	网络探测信号质量回调。
onNetworkTestResult	网络探测详细结果回调。
onUserVolumeStatsNotify	音频音量回调。
onLocalAudioStateChangedNotify	本地音频状态改变回调。
onLocalVideoStateChangedNotify	本地视频状态改变回调。
onNetworkQualityNotify	网络质量上报。
onRenderSuccessNotify	媒体渲染成功上报。
onLocalVolumeChangedNotify	本地采集音量实时回调。
onFirstLocalAudioFrameNotify	本地音频首帧发送回调。

接口	描述
onFirstLocalVideoFrameNotify	本地视频首帧渲染回调。
onMediaConnectStateChangedNotify	媒体服务器连接状态变化回调。
onStartAllRemoteViewResult	批量选看结果回调。
onStatsNotify	当前会话统计回调。
onLocalVideoStatsNotify	本地视频流详情，2s触发一次回调。
onRemoteVideoStatsNotify	远端视频流详情，2s触发一次回调。
onLocalAudioStatsNotify	本地音频流详情，2s触发一次回调。
onRemoteAudioStatsNotify	远端音频流详情，2s触发一次回调。
onVideoResolutionChangedNotify	视频分辨率大小改变回调。
onAudioClipFinished	音效文件播放结束回调。
onAudioFileFinished	音频文件播放结束回调。
onAudioMixStateChangedNotify	混音音频文件播放状态改变回调。
onSeiSendMsgSuccess	音频SEI信息发送成功回调
onSeiRecvMsg	接收音频SEI信息回调
onStartPublishStream	开始旁路（RTMP）推流回调
onUpdateTransCoding	更新旁路（RTMP）推流消息
onStopPublishStream	停止旁路（RTMP）推流消息
onStreamPublishStateChange	RTMP推流状态回调
onAudioDeviceTestVolumeNotify	音频设备测试回调
onShareSourceInfoChangedNotify	共享源目标发生变化通知
onShareWindowLocationChangedNotify	共享源目标坐标发生改变通知
onLocalAudioMutedStatusDetected	本端禁音状态检测回调
onMultiRoomMediaRelayStateChanged	跨房状态改变回调
onRemoteMicrophoneStateChanged	麦克风设备状态变更通知
onUserNetworkQualityNotify	加入房间后的网络质量状态回调
onRoomStreamStatusNotify	房间流状态回调

onError

```
virtual void onError(int error, const char* msg)
```

【功能说明】

发生错误，触发此回调。返回**客户端错误码**或者**服务端错误码**。

【回调参数】

- error：错误码，具体请参见**HRTCErrorCode**。
- msg：错误描述。

onJoinRoomSuccess

```
virtual void onJoinRoomSuccess(const char* roomId, const char* userId)
```

【功能说明】

成功加入房间，触发此回调。

【回调参数】

- roomId：新加入的房间ID。
- userId：新加入房间的用户ID。

onJoinRoomFailure

```
virtual void onJoinRoomFailure(int error, const char* msg)
```

【功能说明】

加入房间失败，触发此回调。

【回调参数】

- error：错误码，具体请参见**HRTCErrorCode**。
- msg：错误描述。

onLeaveRoom

```
virtual void onLeaveRoom(HRTCLeaveReason reason, const HRTCStatsInfo &statsInfo)
```

【功能说明】

离开房间，触发此回调。

【回调参数】

- reason：离开的房间原因，具体请参见**HRTCLeaveReason**。
- statsInfo：卡顿统计信息，具体请参见**HRTCStatsInfo**。

⚠ 注意

APP调用**leaveRoom**接口时，会返回**HRTC_LEAVE_REASON_USER_LEAVE_ROOM**，可以通过以下任一方式回退到登录界面。

- APP在调用**leaveRoom**接口时退到登录界面，或者在收到onLeaveRoom回调，且回调消息不等于**HRTC_LEAVE_REASON_USER_LEAVE_ROOM**时（防止重复操作）退到登录界面。
- APP只在收到onLeaveRoom消息时退到登录界面。

onRemoteUserOnline

```
virtual void onRemoteUserOnline(const char* roomId, const char* userId, const char* userName)
```

【功能说明】

远端joiner用户加入当前房间，触发此回调。该回调提示有远端joiner用户加入了房间，并返回新加入用户的ID。

【回调参数】

- roomId：房间ID。
- userId：远端用户ID。
- userName：远端用户昵称。

onRemoteUserOffline

```
virtual void onRemoteUserOffline(const char* roomId, const char* userId, int reason)
```

【功能说明】

远端joiner用户离开当前房间，触发此回调。

本端用户离开当前房间，会回调当前房间所有用户offline。

【回调参数】

- roomId：当前房间的房间ID。
- userId：离开房间的远端用户ID。
- reason：远端用户离线原因，预留参数。

onRemoteUserNameChangedNotify

```
virtual void onRemoteUserNameChangedNotify(const char* roomId, const char* userId, const char* userName)
```

【功能说明】

远端用户昵称变化，触发此回调。

【回调参数】

- roomId：房间ID。
- userId：用户ID。
- userName：变更后的昵称。

onUserNameChangedNotify

```
virtual void onUserNameChangedNotify(const char* oldUserName, const char* newUserName)
```

【功能说明】

本端用户昵称变化，触发此回调。

【回调参数】

- oldUserName：变更前的昵称。
- newUserName：变更后的昵称。

onFirstRemoteAuxiliaryStreamDecoded

```
virtual void onFirstRemoteAuxiliaryStreamDecoded(const char* roomId, const char* userId, int width, int height)
```

【功能说明】

接收到第一帧远端辅流并解码成功，触发此回调。

【回调参数】

- roomId：视频流对应的房间ID。
- userId：视频流对应的用户ID。
- width：视频流宽。
- height：视频流高。

onFirstRemoteVideoDecoded

```
virtual void onFirstRemoteVideoDecoded(const char* roomId, const char* userId, int width, int height)
```

【功能说明】

接收到第一帧远端视频流并解码成功，触发此回调。

【回调参数】

- roomId：视频流对应的房间ID。
- userId：视频流对应的用户ID。
- width：视频流宽。
- height：视频流高。

onConnectionChangedNotify

```
virtual void onConnectionChangedNotify(HRTCConnStateTypes connType, HRTCConnChangeReason reason, const char* description)
```

【功能说明】

网络连接状态发生变化，触发此回调。

【回调参数】

- connType：网络连接状态。具体请参见**HRTCConnStateTypes**。
- reason：网络连接状态发生变化原因。具体请参见**HRTCConnChangeReason**。
- description：错误原因描述。

onDeviceStateChangedNotify

```
virtual void onDeviceStateChangedNotify(const char* deviceId, HRTCDeviceType deviceType,
HRTCDeviceState deviceState)
```

【功能说明】

设备状态发生变化，触发此回调。

【回调参数】

- deviceId：系统设备标识，如系统音频播放设备标识可通过**getPlaybackDevices**获取。
- deviceType：系统设备类型，具体请参见**HRTCDeviceType**。
- deviceState：系统设备状态，具体请参见**HRTCDeviceState**。

⚠ 注意

通话前插拔设备会上报变化。

onDeviceVolumeChangedNotify

```
virtual void onDeviceVolumeChangedNotify(HRTCDeviceType deviceType, unsigned int volume, unsigned int
muted)
```

【功能说明】

音频设备音量发生变化，触发此回调。

【回调参数】

- deviceType：系统设备类型，具体请参见**HRTCDeviceType**。
- volume：音量
- muted：true表示设备静音，false表示设备未静音。

⚠ 注意

通话前调整音频设备音量和静音会上报变化。

onLogUploadResult

```
virtual void onLogUploadResult(int result)
```

【功能说明】

日志上传结果回调。

【回调参数】

result：日志上传结果。

- 0：上传成功
- >0：上传失败

onLogUploadProgress

```
virtual void onLogUploadProgress(int progress)
```

【功能说明】

日志上传进度回调。

【回调参数】

progress：日志上传进度。取值范围为[0,100]。

onUserRoleChangedNotify

```
virtual void onUserRoleChangedNotify(const char* roomId, HRTCRoleType oldRole, HRTCRoleType newRole)
```

【功能说明】

用户角色切换成功，触发此回调。

【回调参数】

- roomId：发生角色切换的房间号。
- oldRole：切换前的角色。具体请参见**HRTCRoleType**。
- newRole：切换成功后的角色。具体请参见**HRTCRoleType**。

onScreenShareStarted

```
virtual void onScreenShareStarted()
```

【功能说明】

屏幕流共享开启，触发此回调。

onScreenShareStopped

```
virtual void onScreenShareStopped(int reason)
```

【功能说明】

屏幕流共享关闭，触发此回调。

【回调参数】

reason：屏幕共享关闭原因。

onUserAuxiliaryStreamAvailable

```
virtual void onUserAuxiliaryStreamAvailable(const char* roomId, const char* userId, bool available)
```

【功能说明】

远端开启，停止辅流后，触发此回调。

【回调参数】

- roomId：房间ID。
- userId：远端用户ID。
- available：true表示远端开启辅流，false表示远端停止辅流。

onVideoStatsNotify

```
virtual void onVideoStatsNotify(HRTCLocalVideoStats *localStats, unsigned int localStatsCount,
HRTCRemoteVideoStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

视频流详情，2s触发一次回调。

【回调参数】

- localStats：本地视频发流统计，具体请参见**HRTCLocalVideoStats**。
- localStatsCount：localStats数组长度。
- remoteStats：远端视频收流统计，具体请参见**HRTCRemoteVideoStats**。
- remoteStatsCount：remoteStats数组长度。

⚠ 注意

- 当无本地视频时，localStatsCount为0，localStats为空指针，需要先判断不为空再使用，否则可能引发空指针错误。
- 当无远端视频时，remoteStatsCount为0，remoteStats为空指针，需要先判断不为空再使用，否则可能引发空指针错误。

onAudioStatsNotify

```
virtual void onAudioStatsNotify(HRTCLocalAudioStats *localStats, unsigned int localStatsCount,
HRTCRemoteAudioStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

音频流详情，2s触发一次回调。

【回调参数】

- localStats：本地音频发流统计，具体请参见**HRTCLocalAudioStats**。
- localStatsCount：localStats数组长度。
- remoteStats：远端音频收流统计，具体请参见**HRTCRemoteAudioStats**。
- remoteStatsCount：remoteStats数组长度。

⚠ 注意

- 当无本地音频时，localStatsCount为0，localStats为空指针，需要先判断不为空再使用，否则可能引发空指针错误。
- 当无远端音频时，remoteStatsCount为0，remoteStats为空指针，需要先判断不为空再使用，否则可能引发空指针错误。

onAuxiliaryStreamStatsNotify

```
virtual void onAuxiliaryStreamStatsNotify(HRTCLocalVideoStats *localStats, unsigned int localStatsCount,
HRTCRemoteVideoStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

辅流详情，2s触发一次回调。

【回调参数】

- localStats：本地辅流发流统计，具体请参见**HRTCLocalVideoStats**。
- localStatsCount：localStats数组长度。
- remoteStats：远端辅流收流统计，具体请参见**HRTCRemoteVideoStats**。
- remoteStatsCount：remoteStats数组长度。

⚠ 注意

- 当无本地辅流时，localStatsCount为0，localStats为空指针，需要先判断不为空再使用，否则可能引发空指针错误。
- 当无远端辅流时，remoteStatsCount为0，remoteStats为空指针，需要先判断不为空再使用，否则可能引发空指针错误。

onAuthorizationExpired

```
virtual void onAuthorizationExpired();
```

【功能说明】

鉴权签名过期回调，需要app调用更新签名。

onRemoteAudioStateChangedNotify

```
virtual void onRemoteAudioStateChangedNotify(const char* userId, HRTCRemoteAudioStreamState state,
HRTCRemoteAudioStreamStateReason reason)
```

【功能说明】

远端音频流状态变化回调。

【回调参数】

- userId：远端用户ID。
- state：远端音频流状态，具体请参见**HRTCRemoteAudioStreamState**。
- reason：远端音频流状态变化原因，具体请参见**HRTCRemoteAudioStreamStateReason**。

onRemoteVideoStateChangedNotify

```
virtual void onRemoteVideoStateChangedNotify(const char* userId, HRTCRemoteVideoStreamState state,
HRTCRemoteVideoStreamStateReason reason)
```

【功能说明】

远端视频流状态变化回调。

【回调参数】

- userId：远端用户ID。
- state：远端视频流状态，具体请参见**HRTCRemoteVideoStreamState**。
- reason：远端视频流状态变化原因，具体请参见**HRTCRemoteVideoStreamStateReason**。

onRenderAuxiliaryExternalVideoFrame

```
virtual void onRenderAuxiliaryExternalVideoFrame(const char* roomId, HRTCMediaDirection direction, const char* userId, HRTCVideoFrame& videoFrame)
```

【功能说明】

辅流自渲染回调。需要调用setAuxiliaryExternalVideoFrameOutput接口开启辅流自渲染，从而触发该回调。

【回调参数】

- roomId：房间ID。
- direction：数据源，本地数据，远端数据，具体请参见**HRTCMediaDirection**。
- userId：用户ID。
- videoFrame：辅流详情，具体请参见**HRTCVideoFrame**。

onRenderExternalVideoFrame

```
virtual void onRenderExternalVideoFrame(const char* roomId, HRTCMediaDirection direction, const char* userId, HRTCVideoFrame& videoFrame)
```

【功能说明】

视频自渲染回调。需要调用**setExternalVideoFrameOutput**接口开启视频自渲染，从而触发该回调。

【回调参数】

- roomId：房间ID。
- direction：数据源，本地数据，远端数据，具体请参见**HRTCMediaDirection**。
- userId：视频数据对应的远端用户ID。
- videoFrame：视频帧详情，具体请参见**HRTCVideoFrame**。

onPlaybackExternalAudioFrame

```
virtual void onPlaybackExternalAudioFrame(const char* roomId, HRTCMediaDirection direction, HRTCAudioFrame& audioFrame)
```

【功能说明】

音频自渲染回调。需要调用**setExternalAudioFrameOutput**接口开启音频自渲染，从而触发该回调。

【回调参数】

- roomId：房间ID。
- direction：数据源，本地数据，远端数据，具体请参见**HRTCMediaDirection**。
- audioFrame：音频帧详情，具体请参见**HRTCAudioFrame**。

onRejoinRoomSuccess

```
virtual void onRejoinRoomSuccess(const char* roomId, const char* userId)
```

【功能说明】

重新加入房间回调。例如网络异常后重连成功加入房间触发。

【回调参数】

- roomId：房间ID。
- userId：用户ID。

onNetworkTestQuality

```
virtual void onNetworkTestQuality(HRTCNetworkQualityLevel level)
```

【功能说明】

加房间前网络探测回调。

【回调参数】

level：网络质量，具体请参见**HRTCNetworkQualityLevel**。

onNetworkTestResult

```
virtual void onNetworkTestResult(HRTCNetworkTestResult& networkTestResult)
```

【功能说明】

加房间前网络探测结果回调。

【回调参数】

networkTestResult：主要包括测试成功与否、上行和下行的网络带宽、丢包、延时和抖动，具体请参见**HRTCNetworkTestResult**。

onUserVolumeStatsNotify

```
virtual void onUserVolumeStatsNotify(const HRTCVolumeInfo* userVolumes, unsigned int
userVolumesCount, unsigned int totalVolume)
```

【功能说明】

用户音量状态回调。通过**enableUserVolumeNotify**开启并设置回调周期，定时上报。

【回调参数】

- userVolumes：用户信息，具体请参见**HRTCVolumeInfo**。
- userVolumesCount：上报的用户人数，包含本地用户。
- totalVolume：总音量。

onTopActiveSpeaker

```
virtual void onTopActiveSpeaker(const char* userId, bool noStream)
```

【功能说明】

声控画面的用户ID变化时，触发此回调。该回调主要用于0号会场场景（订阅用户id为0的音频）。

【回调参数】

userId：返回当前声控画面的用户ID。

noStream：该用户是否有视频流。

⚠ 注意

0号会场模式下，SDK会持续监测（根据一定时间内用户音量大小）当前最活跃的用户，如果最活跃用户发生变化，则触发此回调并上报当前最活跃的用户userId。

onLocalAudioStateChangedNotify

```
virtual void onLocalAudioStateChangedNotify(HRTCLocalAudioStreamState state,
HRTCLocalAudioStreamStateReason reason)
```

【功能说明】

本地音频状态改变，触发此回调。

【回调参数】

- state：本地音频状态，具体请参见**HRTCLocalAudioStreamState**。
- reason：本地音频状态改变的原因，具体请参见**HRTCLocalAudioStreamStateReason**。

onLocalVideoStateChangedNotify

```
virtual void onLocalVideoStateChangedNotify(HRTCLocalVideoStreamState state,
HRTCLocalVideoStreamStateReason reason)
```

【功能说明】

本地视频状态改变，触发此回调。

【回调参数】

- state：本地视频状态，具体请参见**HRTCLocalVideoStreamState**。
- reason：本地视频状态改变原因，具体请参见**HRTCLocalVideoStreamStateReason**。

onNetworkQualityNotify

```
virtual void onNetworkQualityNotify(HRTCQualityInfo* localQuality, unsigned int localQualityCount,
HRTCQualityInfo* remoteQuality, unsigned int remoteQualityCount)
```

【功能说明】

房间内客户端基于流级别的网络质量实时上报，默认开启，每2s上报一次，有数据流时才会回调，音频流、视频流分开回调。

【回调参数】

- localQuality：本地上行网络质量，该参数暂时不使用。
- localQualityCount：正在上报的网络质量数量，该参数暂时不使用。
- remoteQuality：（本地下行）远端各路流的网络质量，具体请参见**HRTCQualityInfo**。
- remoteQualityCount：正在上报的流的数量，集合的大小。

onRenderSuccessNotify

```
virtual void onRenderSuccessNotify(const char* userId, unsigned int isAux)
```

【功能说明】

媒体恢复渲染成功回调。

回调包括以下场景：

- 视频流第一次渲染；
- 分辨率有变化；
- 一段时间没有渲染后恢复渲染（2s）。

【回调参数】

- userId：用户ID。
- isAux：是否是辅流。

onLocalVolumeChangedNotify

```
virtual void onLocalVolumeChangedNotify(unsigned int volume, unsigned int muted)
```

【功能说明】

本地采集音量回调。

【回调参数】

- volume：麦克风采集音量。
- muted：是否静音，用于区分是没有声音还是muted。

onFirstLocalAudioFrameNotify

```
virtual void onFirstLocalAudioFrameNotify(unsigned int elapsed)
```

【功能说明】

本地音频首帧发送回调。

【回调参数】

elapsed：从入会到本地音频首帧发送所用的时间，单位ms。

onFirstLocalVideoFrameNotify

```
virtual void onFirstLocalVideoFrameNotify(unsigned int elapsed)
```

【功能说明】

本地视频首帧渲染回调。

【回调参数】

elapsed：从开始采集到本地视频首帧渲染所用的时间，单位ms。

onMediaConnectStateChangedNotify

```
virtual void onMediaConnectStateChangedNotify(HRTCMediaConnStateTypes state,
HRTCMediaConnChangeReason reason, const char* description)
```

【功能说明】

媒体服务器连接状态变更通知。

【回调参数】

- state：与媒体服务器连接状态，具体请参见**HRTCMediaConnStateTypes**。
- reason：连接状态变化的原因，具体请参见**HRTCMediaConnChangeReason**。
- description：连接状态变化原因描述。

⚠ 注意

加入房间过后，收到媒体服务的数据包时，返回Connected消息，超过6s没有收到包，则返回Failed消息。

onStartAllRemoteViewResult

```
virtual void onStartAllRemoteViewResult(int errCode, const char* errMsg, unsigned int counts, const HRTCSetupRemoteViewResult* results)
```

【功能说明】

批量选看结果回调。

【回调参数】

- errCode：错误码。
- errMsg：错误信息。
- counts：results数组大小。
- results：批量选看结果，具体请参见**HRTCSetupRemoteViewResult**。

onStatsNotify

```
virtual void onStatsNotify(HRTCOnStats *rtcStats)
```

【功能说明】

当前会话统计回调。

【回调参数】

rtcStats：当前会话统计，具体请参见**HRTCOnStats**。

onLocalVideoStatsNotify

```
virtual void onLocalVideoStatsNotify(const HRTCLocalVideoStats *localStats, unsigned int localStatsCount)
```

【功能说明】

本地视频流详情，2s触发一次回调。

【回调参数】

- localStats：本地视频收流统计，具体请参见**HRTCLocalVideoStats**。
- localStatsCount：localStats数组长度。

onRemoteVideoStatsNotify

```
virtual void onRemoteVideoStatsNotify(const HRTCRemoteVideoStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

远端视频流详情，2s触发一次回调。

【回调参数】

- remoteStats：远端视频收流统计，具体请参见**HRTCRemoteVideoStats**。
- remoteStatsCount：remoteStats数组长度。

onLocalAudioStatsNotify

```
virtual void onLocalAudioStatsNotify(const HRTCLocalAudioStats *localStats, unsigned int localStatsCount)
```

【功能说明】

本地音频流详情，2s触发一次回调。

【回调参数】

- localStats：本地音频收流统计，具体请参见**HRTCLocalAudioStats**。
- localStatsCount：localStats数组长度。

onRemoteAudioStatsNotify

```
virtual void onRemoteAudioStatsNotify(const HRTCRemoteAudioStats *remoteStats, unsigned int
remoteStatsCount)
```

【功能说明】

远端音频流详情，2s触发一次回调。

【回调参数】

- remoteStats：远端音频收流统计，具体请参见**HRTCRemoteAudioStats**。
- remoteStatsCount：remoteStats数组长度。

onVideoResolutionChangedNotify

```
virtual void onVideoResolutionChangedNotify(const char* userId, int width, int height)
```

【功能说明】

远端视频分辨率大小改变，触发此回调。

【回调参数】

- userId：用户ID。
- width：视频分辨率改变后的宽。
- height：视频分辨率改变后的高。

onAudioClipFinished

```
virtual void onAudioClipFinished(int soundId)
```

【功能说明】

音效文件播放结束，触发此回调。

【回调参数】

soundId：音效ID，取值>=0。

onAudioFileFinished

```
virtual void onAudioFileFinished()
```

【功能说明】

音频文件播放结束，触发此回调。

【回调参数】

无

onAudioMixStateChangedNotify

```
virtual void onAudioMixStateChangedNotify(HRTCAudioFileState state, HRTCAudioFileReason reason,
unsigned long long value)
```

【功能说明】

音频文件播放状态改变，触发此回调。

【回调参数】

- state：音频播放状态，具体请参见**HRTCAudioFileState**。
- reason：音频播放状态改变原因，具体请参见**HRTCAudioFileReason**。
- value：state为HWRtcAudioFileOpenCompleted表示音频文件的时长，单位为ms；state为HWRtcAudioFilePositionUpdate表示当前播放的位置，单位为ms。其他情况下，value值无意义。

onSeiSendMsgSuccess

```
void onSeiSendMsgSuccess(const char* message);
```

【功能说明】

音频SEI信息发送成功回调。

【回调参数】

message：发送SEI信息的内容。

onSeiRecvMsg

```
void onSeiRecvMsg(const char* userId, const char* message);
```

【功能说明】

接收音频SEI信息回调。

【回调参数】

- userId：用户ID。
- message：接收SEI信息的内容。

onStartPublishStream

```
void onStartPublishStream(int code, const char* taskId);
```

【功能说明】

开始旁路（RTMP）推流回调。

【回调参数】

- code：错误码，成功为0，失败请参考错误码**HRTCErrorCode**。
- taskId：任务Id。

onUpdateTransCoding

```
void onUpdateTransCoding(int code, const char* taskId);
```

【功能说明】

更新旁路（RTMP）推流消息。

【回调参数】

- code：错误码，成功为0，失败请参考错误码**HRTCErrorCode**。
- taskId：任务Id。

onStopPublishStream

```
void onStopPublishStream(int code, const char* taskId);
```

【功能说明】

停止旁路（RTMP）推流消息。

【回调参数】

- code：错误码，成功为0，失败请参考错误码**HRTCErrorCode**。
- taskId：任务Id。

onStreamPublishStateChange

```
void onStreamPublishStateChange(int code, const char* taskId, const HRTCUrlStatusList * urlStatu);
```

【功能说明】

RTMP推流状态回调。

【回调参数】

- code：错误码，成功为0，失败请参考错误码**HRTCErrorCode**。
- taskId：任务Id。
- urlStatu：推流的url状态，具体请参见**HRTCUrlStatusList**。

onAudioDeviceTestVolumeNotify

```
void onAudioDeviceTestVolumeNotify(HRTCAudioDeviceTestVolumeNotify *data);
```

【功能说明】

音频设备测试回调。

【回调参数】

data：回调数据，具体请参见**HRTCAudioDeviceTestVolumeNotify**。

onShareSourceInfoChangedNotify

```
void onShareSourceInfoChangedNotify(HRTCShareSourceInfoChangedType type);
```


【功能说明】

共享源目标发生变化通知。

【回调参数】

type：具体请参见**HRTCShareSourceInfoChangedType**

onShareWindowLocationChangedNotify

```
void onShareWindowLocationChangedNotify(const char* info);
```

【功能说明】

共享源目标坐标或者DPI发生变化通知。

【回调参数】

info：json类型，如{dpi:100, x:0, y:0, width:1920, height:1080}

- dpi：当前窗口的缩放
- x：左上角x轴坐标
- y：左上角y轴坐标
- width：宽
- height：高

onLocalAudioMutedStatusDetected

```
void onLocalAudioMutedStatusDetected();
```

【功能说明】

本端禁音状态被检测到。

回调参数

无

⚠ 注意

通过**enableUserVolumeNotify**开启并设置回调周期，本端禁音后检测到麦克风有输入后定时上报。上报频率和**enableUserVolumeNotify**的参数大小相关，参考值建议设置成200。

onMultiRoomMediaRelayStateChanged

```
void onMultiRoomMediaRelayStateChanged(const char *roomId, HRTCMultiRoomMediaRelayState state,
HRTCMultiRoomMediaRelayStateCode code);
```

【功能说明】

跨房状态回调。

【回调参数】

roomId：跨房房间号。

state：状态类型，具体请参见**HRTCMultiRoomMediaRelayState**。

code：状态的具体原因，具体请参见**HRTCMultiRoomMediaRelayStateCode**。

onRemoteMicrophoneStateChanged

```
void onRemoteMicrophoneStateChanged(const char* userId, HRTCRemoteMicState state);
```

【功能说明】

远端麦克风设备状态变更通知。

【回调参数】

userId：远端用户userId。

state：麦克风设备状态，具体请参见**HRTCRemoteMicState**。

onUserNetworkQualityNotify

```
void onUserNetworkQualityNotify(const char *roomId, const char* userId, HRTCNetworkQualityLevel
txQuality,  HRTCNetworkQualityLevel rxQuality);
```

【功能说明】

支持用户上报各自与近端SFU间的上下行网络质量，基于用户级，使本地用户能获取同房间内远端用户与其近端SFU间的网络质量。CMD广播时为房间级，将广播给订阅了此主播流的用户或者此主播被选为TOPN用户且观众订阅了该TOPN用户。

【回调参数】

- roomId：用户所在房间号。
- userId: 上报的用户id，0为本地，非0为远端。
- txQuality：该用户的上行网络质量，具体请参见**HRTCNetworkQualityLevel**。
- rxQuality：该用户的下行网络质量，具体请参见**HRTCNetworkQualityLevel**。

⚠ 注意

- 此接口不支持跨房场景、WebRTC场景。
- 不支持RTSA。

onRoomStreamStatusNotify

```
void onRoomStreamStatusNotify(int audienceState)
```

【功能说明】

房间流状态通知，业务调用云侧暂停/恢复接口后，端侧收到该通知。

【回调参数】

audienceState：0表示暂停 ，1表示恢复。

7.4.3 IHRTCConnection

7.4.3.1 接口总览

本章节介绍了全平台C++ SDK的IHRTCConnection接口详情。

初始化等基础接口

表 7-11 初始化等基础接口

接口	描述
release	释放IHRTCConnection对象资源
getRoomId	获取当前连接房间号
setNetworkBandwidth	设置网络带宽限制

房间功能

表 7-12 房间功能接口

接口	描述
joinRoom	加入房间
leaveRoom	离开房间
renewAuthorization	签名更新
changeUserRole	设置用户的角色，切换角色时使用
changeUserName	设置用户自己的昵称

视频管理

表 7-13 视频管理接口

接口	描述
setupRemoteView	设置远端流渲染视图
updateRemoteRenderMode	设置远端用户视图渲染模式,镜像模式
setRemoteVideoStreamType	大小流模式，设置远端视频流类型
setPriorRemoteVideoStreamType	大小流模式，设置所有订阅的远端视频流类型
pullRemoteVideo	设置是否接收对应远端用户的视频流
pullAllRemoteVideo	设置是否接收所有用户的视频流
startRemoteStreamView	设置远端用户渲染视图（发起选看-老接口）
stopRemoteStreamView	关闭远端用户的渲染视图
setRemoteVideoAdjustResolution	设置是否开启远端分辨率自适应

音频管理

表 7-14 音频管理接口

接口	描述
muteRemoteAudio	设置是否接收指定远端用户的音频流
muteAllRemoteAudio	设置是否接收所有用户的音频流

辅流管理

表 7-15 辅流管理接口

接口	描述
setRemoteAuxiliaryStreamViewRotation	设置远端辅流视图旋转角度
startRemoteAuxiliaryStreamView	开启辅流渲染视图（发起辅流选看）
stopRemoteAuxiliaryStreamView	关闭辅流渲染视图（停止辅流选看）
updateRemoteAuxiliaryStreamRenderMode	设置辅流视图显示模式，镜像模式
setRemoteAuxiliaryStreamViewOrientation	设置远端辅流视图方向（横竖屏）

7.4.3.2 初始化等基础接口

release

```
virtual void release()
```

【功能说明】

释放IHRTCConnection对象，如果未离会，会自动离会再释放连接，会触发onDestroyConnection回调。

⚠ 注意

如果资源已经释放，不能继续使用该对象，需要重新调用createConnection。

getRoomId

```
virtual const char* getRoomId()
```

【功能说明】

获取当前连接对应的房间ID。

【请求参数】

无

【返回参数】

roomId：返回当前连接对应的房间ID。

setNetworkBandwidth

```
virtual int setNetworkBandwidth(const HRTCNetworkBandwidth &bandwidthParam)
```

【功能说明】

设置网络带宽限制，在每次加入房间之前调用。

【请求参数】

bandwidthParam：带宽设置参数，具体请参见**HRTCNetworkBandwidth**。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HRTCErrorCode**。

7.4.3.3 房间功能

joinRoom

```
virtual int joinRoom(const HRTCJoinParam &joinParam)
```

【功能说明】

加入房间。该方法让用户加入通话房间。

【请求参数】

joinParam：入会参数。具体请参见**HRTCJoinParam**。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

会触发以下回调：

- **onConnectStateChange**：连接状态发送改变。
- **onJoinRoomSuccess**：加入房间成功。
- **onJoinRoomFailure**：加入房间失败。
- **onRemoteUserOnline**：远端用户收到当前用户加入房间的通知。

音频的自动订阅策略设置只在音频订阅模式下生效。

leaveRoom

```
virtual int leaveRoom()
```

【功能说明】

离开房间。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

会触发以下回调：

- **onLeaveRoom**：离开房间回调。
- **onConnectStateChange**：连接状态改变回调。
- **onRemoteUserOffline**：远端用户收到当前用户离开房间的通知。

renewAuthorization

```
virtual int renewAuthorization(const char* signature, long long ctime)
```

【功能说明】

鉴权签名过期，收到**onAuthorizationExpired**签名鉴权过期回调后，更新鉴权签名。

【请求参数】

- signature：鉴权签名字符串。
- ctime：过期时间。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

changeUserRole

```
virtual int changeUserRole(HRTCRoleType role, const char *authorization, long long ctime)
```

【功能说明】

设置用户在当前房间内的角色类型，角色切换时使用。

【请求参数】

- role：用户角色类型，joiner类型和player类型，具体请参见**HRTCErrorCode**。
- authorization：预留参数，填null。
- ctime：预留参数，填0。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 加入房间前，可以通过joinRoom的userRole参数确认角色信息。
- 加入指定房间后才可以在指定房间内进行角色切换，当前仅支持joiner和player角色切换。跨房场景下，通过对应connection连接下的changeUserRole接口实现在跨入房间中的角色类型切换。
- 切换成功触发**onUserRoleChangedNotify**回调。切换失败会触发**onError**回调，返回HRTC_ERR_CODE_USER_ROLE_CHANGE_FAIL错误码。
- 同一时间不同房间最多只能有一个joiner，player切换joiner的时候，需要将其他房间的joiner先切换成player。
- 不支持缺省用户昵称入会。

changeUserName

```
virtual int changeUserName(const char* userName)
```

【功能说明】

房间内设置用户自己的昵称。

【请求参数】

userName：变更的昵称。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 该接口仅支持房间内调用，更改的昵称会被实时同步到房间内其他用户的用户列表，退出房间不会保存，再次加入房间变更为加入房间时设置的昵称（参考**joinRoom**接口注意事项）。
- 会触发以下回调：

  onUserNameChangedNotify：用户名变更的通知。

7.4.3.4 音频管理

muteRemoteAudio

```
virtual int muteRemoteAudio(const char* userId, bool mute)
```

【功能说明】

设置是否接收对应远端用户的音频流。同一时间所有房间最多只能接收17路音频流。

【请求参数】

- userId: 远端用户的userId，唯一标识。
- mute：true表示取消音频流接收。false表示开启音频流接收。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

muteAllRemoteAudio

```
virtual int muteAllRemoteAudio(bool mute)
```

【功能说明】

设置是否接收当前房间所有用户的音频流。

【请求参数】

mute：true表示取消接收，false表示开启接收。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 取消所有音频流接收，同时也会取消接收新加入用户的音频流。
- 开启所有音频流接收，同时也会开启接收新加入用户的音频流。
- 默认开启所有音频流接收。
- 不支持音频订阅模式。

7.4.3.5 视频管理

setupRemoteView

```
virtual int setupRemoteView(const char* userId, view_t view)
```

【功能说明】

设置远端流渲染视图（新选看接口），该接口不影响收流。

【请求参数】

- userId：远端用户的唯一标识。
- view：窗口句柄，view为NULL时，解除窗口绑定并结束选看。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

updateRemoteRenderMode

```
virtual int updateRemoteRenderMode(const char* userId, HRTCVideoDisplayMode displayMode,
HRTCVideoMirrorType mirrorMode)
```

【功能说明】

设置远端用户视图渲染模式。

【请求参数】

- userId：远端用户的唯一标识。
- displayMode：视图显示模式。具体请参见**HRTCVideoDisplayMode**，默认RTC_VIDEO_DISPLAY_HIDDEN，通过裁剪的方式保持宽高比。
- mirrorMode：镜像模式。具体请参见**HRTCVideoMirrorType** 。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setRemoteVideoStreamType

```
virtual int setRemoteVideoStreamType(const char *userId, HRTCVideoStreamType type)
```

【功能说明】

大小流模式，设置指定选看用户的视频流类型。在通过新选看接口发起选看时调用。

【请求参数】

- userId：远端用户唯一标识。
- type：视频流类型。指大流、小流，具体请参见**HRTCVideoStreamType**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setPriorRemoteVideoStreamType

```
virtual int setPriorRemoteVideoStreamType(HRTCVideoStreamType type)
```

【功能说明】

大小流模式，设置所有订阅的远端视频流类型。默认订阅大流，优先使用setRemoteVideoStreamType接口设置的用户流类型。

【请求参数】

type：视频流类型。指大流、小流，具体请参见**HRTCVideoStreamType**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

pullRemoteVideo

```
virtual int pullRemoteVideo(const char* userId, bool pull)
```

【功能说明】

开启/关闭接收指定远端用户的视频流。只能加入房间后调用。

【请求参数】

- userId: 远端用户的userId，唯一标识。
- pull：true表示开启接收，false表示关闭接收。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

pullAllRemoteVideo

```
virtual int pullAllRemoteVideo(bool mute)
```

【功能说明】

开启/关闭接收当前房间所有远端用户的视频流。

【请求参数】

mute：true表示关闭接收，false表示开启接收， 默认开启接收。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

startRemoteStreamView

```
virtual int startRemoteStreamView(const char* userId, view_t view, HRTCStreamType streamType, bool disableAdjustRes)
```

【功能说明】

设置远端用户渲染视图，并开启收流（选看）。

【请求参数】

- userId：远端用户的唯一标识。
- view：窗口句柄。
- streamType：视频分辨率，具体请参见**HRTCStreamType**。
- disableAdjustRes：禁用分辨率自适应，默认值false，开启分辨率自适应。若关闭，在网络环境较差情况下可能会有卡顿现象。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

注意

该接口为选看的旧接口，通过该接口和updateRemoteRenderMode完成一次完整的选看流程。新的完成选看功能拆分为三个接口：**setupRemoteView**、**pullRemoteVideo**和**setRemoteVideoStreamType**接口，将设置渲染模式、窗口句柄、选看的流类型拆分并增加pullRemoteVideo收流控制接口，以实现更细化的选看流程控制（将窗口绑定和收流控制分开）。您可以根据需要选择调用不同的接口组合以实现视频选看。

stopRemoteStreamView

```
virtual int stopRemoteStreamView(const char* userId)
```

【功能说明】

关闭远端用户的渲染视图，并停止收流（停止选看）。

【请求参数】

userId：远端用户的唯一标识。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

注意

新选看建议通过**setupRemoteView**接口设置窗口句柄为空来停止选看。

setRemoteVideoAdjustResolution

```
virtual int setRemoteVideoAdjustResolution(bool enable)
```

【功能说明】

设置是否开启远端流分辨率自适应。默认开启自适应。

【请求参数】

enable：是否开启自适应。默认开启。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

7.4.3.6 辅流管理

setRemoteAuxiliaryStreamViewRotation

```
virtual int setRemoteAuxiliaryStreamViewRotation(const char *userId, HRTCVideoRotation rotation)
```

【功能说明】

设置远端辅流视图旋转角度。

【请求参数】

- userId：用户ID。
- rotation：旋转角度信息（0°，90°，270°），具体请参见**HRTCVideoRotation**。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HRTCErrorCode**。

setRemoteAuxiliaryStreamViewOrientation

```
virtual int setRemoteAuxiliaryStreamViewOrientation(const char *userId, HRTCVideoOrientation orientation)
```

【功能说明】

设置远端辅流视图方向（横竖屏）。

【请求参数】

- userId：用户ID。
- orientation：方向（横竖屏），具体请参见**HRTCVideoOrientation**。

【返回参数】

- 0：方法调用成功。
- >0：方法调用失败。具体请参见**HRTCErrorCode**。

startRemoteAuxiliaryStreamView

```
virtual int startRemoteAuxiliaryStreamView(const char* userId, view_t view)
```

【功能说明】

当远端开启辅流，本地接收到远端辅流开启**onUserAuxiliaryStreamAvailable**消息后，设置辅流流窗口视图（发起辅流选看）。

【请求参数】

- userId：远端用户的唯一标识。
- view ： 窗口句柄。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 收到**onUserAuxiliaryStreamAvailable**消息后，获取对应的userId。
- 多辅流场景，一个用户同时只能订阅一条辅流；当前正在订阅用户A的辅流，需要订阅另一个用户B的辅流时，需要先停止订阅用户A的辅流，再订阅用户B的辅流。

stopRemoteAuxiliaryStreamView

```
virtual int stopRemoteAuxiliaryStreamView(const char* userId)
```

【功能说明】

关闭屏幕辅流窗口视图（停止辅流选看）。

【请求参数】

userId：远端用户的唯一标识。对应**onUserAuxiliaryStreamAvailable**返回的共享用户标识。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

收到**onUserAuxiliaryStreamAvailable**消息后，如果选看的远端辅流不可用，则必须调用**stopRemoteAuxiliaryStreamView**关闭。

updateRemoteAuxiliaryStreamRenderMode

```
virtual int updateRemoteAuxiliaryStreamRenderMode(const char* userId, HRTCVideoDisplayMode displayMode, HRTCVideoMirrorType mirrorMode)
```

【功能说明】

设置辅流视图渲染模式。

【请求参数】

- userId：远端用户的唯一标识。
- displayMode：视图显示模式 。具体请参见**HRTCVideoDisplayMode**，默认RTC_VIDEO_DISPLAY_FIT，通过扩边的方式保持宽高比。
- mirrorMode：镜像模式。具体请参见**HRTCVideoMirrorType**。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

7.4.4 事件回调（IHRTCConnection）

本章节介绍了全平台C++ SDK的回调接口IHRTCEngineEventHandler的详情。

表 7-16 回调

接口	描述
onError	错误回调。
onWarning	警告回调。
onConnectionChangedNotify	连接状态改变回调。
onAuthorizationExpired	鉴权签名过期回调。
onJoinRoomSuccess	成功加入房间回调。
onJoinRoomFailure	加入房间失败回调。
onRejoinRoomSuccess	重新加入房间回调。
onLeaveRoom	离开房间回调。
onUserRoleChangedNotify	用户角色切换成功回调。
onRemoteUserOnline	远端用户加入当前房间回调。
onRemoteUserOffline	远端用户离开当前房间回调。
onRemoteUserNameChangedNotify	远端用户昵称变化回调。
onUserNameChangedNotify	本地用户昵称变化回调。
onRemoteAudioStateChangedNotify	远端音频流状态变化回调。
onRemoteVideoStateChangedNotify	远端视频流状态变化回调。
onAudioStatsNotify	音频订阅状态回调。
onUserAuxiliaryStreamAvailable	远端开启/停止辅流回调。
onFirstRemoteVideoDecoded	引擎收到第一帧远端视频流并解码成功回调。
onFirstRemoteAuxiliaryStreamDecoded	引擎收到第一帧远端辅流并解码成功回调。
onRenderSuccessNotify	媒体渲染成功上报。
onAudioStatsNotify	音频流详情，2s触发一次回调。
onVideoStatsNotify	视频流详情，2s触发一次回调。
onAuxiliaryStreamStatsNotify	辅流详情，2s触发一次回调。
onNetworkQualityNotify	网络质量上报。
onDestroyConnection	销毁连接。
onMediaConnectStateChangedNotify	媒体服务器连接状态变化回调。
onStartAllRemoteViewResult	批量选看结果回调。

接口	描述
onStatsNotify	当前会话统计回调。
onVideoResolutionChangedNotify	视频分辨率大小改变对调。
onRemoteAudioStatsNotify	远端音频流状态，2秒触发一次回调。
onRemoteVideoStatsNotify	远端视频流状态，2秒触发一次回调。
onStartPublishStream	开始旁路（RTMP）推流回调
onUpdateTransCoding	更新旁路（RTMP）推流回调
onStopPublishStream	停止旁路（RTMP）推流回调
onStreamPublishStateChange	RTMP推流状态回调
onMultiRoomMediaRelayStateChanged	跨房状态改变回调
onRemoteMicrophoneStateChanged	麦克风设备状态变更通知
onRoomStreamStatusNotify	房间流状态回调

onError

```
virtual void onError(IHRTCConnection* conn, int error, const char* msg)
```

【功能说明】

发生错误，触发此回调。返回客户端错误码或者服务端错误码。

【回调参数】

- conn：连接对象。
- error：错误码，具体请参见HRTCErrorCode。
- msg：错误描述。

onWarning

```
virtual void onWarning(IHRTCConnection* conn, int warn, const char* msg)
```

【功能说明】

发生错误，触发此回调。返回客户端错误码或者服务端错误码。

【回调参数】

- conn：连接对象。
- warn：警告码。
- msg：警告描述。

onConnectionChangedNotify

```
virtual void onConnectionChangedNotify(IHRTCConnection* conn, HRTCConnStateType state,
HRTCConnChangeReason reason, const char* description)
```

【功能说明】

连接状态改变回调。

【回调参数】

- conn：连接对象。
- state：连接状态类型，具体请参见**HRTCConnStateTypes**。
- reason：连接状态改变原因，具体请参见**HRTCConnChangeReason**。
- description：连接状态改变描述。

onAuthorizationExpired

```
virtual void onAuthorizationExpired(IHRTCConnection* conn);
```

【功能说明】

鉴权签名过期回调，需要app调用renewAuthorization更新签名。

【回调参数】

conn：连接对象。

onJoinRoomSuccess

```
virtual void onJoinRoomSuccess(IHRTCConnection* conn, const char* userId)
```

【功能说明】

成功加入房间，触发此回调。

【回调参数】

- conn：连接对象。
- userId：新加入房间的用户ID。

onJoinRoomFailure

```
virtual void onJoinRoomFailure(IHRTCConnection* conn, int error, const char* msg)
```

【功能说明】

加入房间失败，触发此回调。

【回调参数】

- conn：连接对象。
- error：错误码，具体请参见**HRTCErrorCode**。
- msg：错误描述。

onRejoinRoomSuccess

```
virtual void onRejoinRoomSuccess(IHRTCConnection* conn, const char* userId)
```

【功能说明】

重新加入房间回调。例如网络异常后重连成功加入房间触发。

【回调参数】

- conn：连接对象。
- userId：用户ID。

onLeaveRoom

```
virtual void onLeaveRoom(IHRTCConnection* conn, HRTCLeaveReason reason, const HRTCStatsInfo &statsInfo)
```

【功能说明】

离开房间，触发此回调。

【回调参数】

- conn：连接对象。
- reason：离开的房间原因，具体请参见**HRTCLeaveReason**。
- statsInfo：卡顿统计信息，具体请参见**HRTCStatsInfo**。

onUserRoleChangedNotify

```
virtual void onUserRoleChangedNotify(IHRTCConnection* conn, HRTCRoleType oldRole, HRTCRoleType newRole)
```

【功能说明】

用户角色切换成功，触发此回调。

【回调参数】

- conn：连接对象。
- oldRole：切换前的角色。具体请参见**HRTCRoleType**。
- newRole：切换成功后的角色。具体请参见**HRTCRoleType**。

onRemoteUserOnline

```
virtual void onRemoteUserOnline(IHRTCConnection* conn, const char* userId, const char* userName)
```

【功能说明】

远端joiner用户加入当前房间，触发此回调。该回调提示有远端joiner用户加入了房间，并返回新加入用户的ID。

【回调参数】

- conn：连接对象。
- userId：远端用户ID。
- userName：远端用户昵称。

onRemoteUserOffline

```
virtual void onRemoteUserOffline(IHRTCConnection* conn, const char* userId, int reason)
```

【功能说明】

远端joiner用户离开当前房间，触发此回调。

本端用户离开当前房间，会回调当前房间所有用户offline。

【回调参数】

- conn：连接对象。
- userId：离开房间的远端用户ID。
- reason：远端用户离线原因，预留参数。

onRemoteUserNameChangedNotify

```
virtual void onRemoteUserNameChangedNotify(IHRTCConnection* conn, const char* userId, const char* userName)
```

【功能说明】

远端用户昵称变化，触发此回调。

【回调参数】

- conn：连接对象。
- userId：用户ID。
- userName：变更后的昵称。

onUserNameChangedNotify

```
virtual void onUserNameChangedNotify(IHRTCConnection* conn, const char* oldUserName, const char* newUserName)
```

【功能说明】

本端用户昵称变化，触发此回调。

【回调参数】

- conn：连接对象。
- oldUserName：变更前的昵称。
- newUserName：变更后的昵称。

onRemoteAudioStateChangedNoitfy

```
virtual void onRemoteAudioStateChangedNotify(IHRTCConnection* conn, const char* userId, HRTCRemoteAudioStreamState state, HRTCRemoteAudioStreamStateReason reason)
```

【功能说明】

远端音频流状态变化回调。

【回调参数】

- conn：连接对象。
- userId：远端用户ID。
- state：远端音频流状态，具体请参见**HRTCRemoteAudioStreamState**。
- reason：远端音频流状态变化原因，具体请参见**HRTCRemoteAudioStreamStateReason**。

onRemoteVideoStateChangedNotify

```
virtual void onRemoteVideoStateChangedNotify(IHRTCConnection* conn, const char* userId, HRTCRemoteVideoStreamState state, HRTCRemoteVideoStreamStateReason reason)
```

【功能说明】

远端视频流状态变化回调。

【回调参数】

- conn：连接对象。
- userId：远端用户ID。
- state：远端视频流状态，具体请参见**HRTCRemoteVideoStreamState**。
- reason：远端视频流状态变化原因，具体请参见**HRTCRemoteVideoStreamStateReason**。

onUserVolumeStatsNotify

```
virtual void onUserVolumeStatsNotify(IHRTCConnection* conn, const HRTCVolumeInfo* userVolumes, unsigned int userVolumesCount, unsigned int totalVolume)
```

【功能说明】

用户音量状态回调。通过**enableUserVolumeNotify**开启并设置回调周期，定时上报。

【回调参数】

- conn：连接对象。
- userVolumes：HRTCVolumeInfo，userId，volume。
- userVolumesCount：音量上报用户数组的大小。
- totalVolume：总音量。

onUserAuxiliaryStreamAvailable

```
virtual void onUserAuxiliaryStreamAvailable(IHRTCConnection* conn, const char* userId, bool available)
```

【功能说明】

远端开启，停止辅流后，触发此回调。

【回调参数】

- conn：连接对象。
- userId：远端用户ID。
- available：true表示远端推辅流，false表示远端停止辅流。

onFirstRemoteVideoDecoded

```
virtual void onFirstRemoteVideoDecoded(IHRTCConnection* conn, const char* userId, int width, int height)
```

【功能说明】

远端用户视频流第一帧解码成功，触发此回调。

【回调参数】

- conn：连接对象。
- userId：用户ID。
- width：视频宽。
- height：视频高。

onFirstRemoteAuxiliaryStreamDecoded

```
virtual void onFirstRemoteAuxiliaryStreamDecoded(IHRTCConnection* conn, const char* userId, int width, int height)
```

【功能说明】

远端用户辅流第一帧解码成功，触发此回调。

【回调参数】

- conn：连接对象。
- userId：用户ID。
- width：视频宽。
- height：视频高。

onRenderSuccessNotify

```
virtual void onRenderSuccessNotify(IHRTCConnection* conn, const char* userId, unsigned int isAux)
```

【功能说明】

媒体渲染成功回调，当前该数据主要用作无码流提示场景中从无码流状态恢复为有码流状态的依据。

【回调参数】

- conn：连接对象。
- userId：用户ID。
- isAux：是否是辅流。

onAudioStatsNotify

```
virtual void onAudioStatsNotify(IHRTCConnection* conn, HRTCLocalAudioStats *localStats, unsigned int localStatsCount, HRTCRemoteAudioStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

音频流详情，2s触发一次回调。

【回调参数】

- conn：连接对象。
- localStats：本地音频发流统计，具体请参见**HRTCLocalAudioStats**。
- localStatsCount：localStats数组长度。
- remoteStats：远端音频收流详情，具体请参见**HRTCRemoteAudioStats**。
- remoteStatsCount：remoteStats数组长度。

⚠ 注意

- 当无本地音频时，localStatsCount为0，localStats为空指针。
- 当无远端音频时，remoteStatsCount为0，remoteStats为空指针。

onVideoStatsNotify

```
virtual void onVideoStatsNotify(IHRTCConnection* conn, HRTCLocalVideoStats *localStats, unsigned int localStatsCount, HRTCRemoteVideoStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

视频流详情，2s触发一次回调。

【回调参数】

- conn：连接对象。
- localStats：本地视频发流统计，具体请参见**HRTCLocalVideoStats**。
- localStatsCount：localStats数组长度。
- remoteStats：远端视频收流统计，具体请参见**HRTCRemoteVideoStats**。
- remoteStatsCount：remoteStats数组长度。

⚠ 注意

- 当无本地视频时，localStatsCount为0，localStats为空指针。
- 当无远端视频时，remoteStatsCount为0，remoteStats为空指针。

onAuxiliaryStreamStatsNotify

```
virtual void onAuxiliaryStreamStatsNotify(IHRTCConnection* conn, HRTCLocalVideoStats *localStats, unsigned int localStatsCount, HRTCRemoteVideoStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

辅流详情，2s触发一次回调。

【回调参数】

- conn：连接对象。
- localStats：本地辅流发流详情，具体请参见**HRTCLocalVideoStats**。
- localStatsCount：本地辅流发流数量。
- remoteStats：远端辅流收流详情，具体请参见**HRTCRemoteVideoStats**。
- remoteStatsCount：远端辅流收流数量。

onNetworkQualityNotify

```
virtual void onNetworkQualityNotify(IHRTCConnection* conn, HRTCQualityInfo* localQuality, unsigned int localQualityCount, HRTCQualityInfo* remoteQuality, unsigned int remoteQualityCount)
```

【功能说明】

房间内客户端基于流级别的网络质量实时上报，默认开启，每2s上报一次，有数据流时才会回调，音频流、视频流分开回调。

【回调参数】

- conn：连接对象。
- localQuality：本地上行网络质量，该参数暂时不使用。
- localQualityCount：正在上报的网络质量数量，该参数暂时不使用。

- remoteQualityCount：正在上报的流的数量，集合的大小。

onDestroyConnection

```
virtual void onDestroyConnection(IHRTCConnection* conn)
```

【功能说明】

调用release释放连接后，触发该回调，这是IHRTCConnection对象最后一个回调，需要在该回调触发后再去清理IHRTCConnectionEventHandler资源。

【回调参数】

conn：连接对象。

onMediaConnectStateChangedNotify

```
virtual void onMediaConnectStateChangedNotify(IHRTCConnection* conn, HRTCMediaConnStateTypes state, HRTCMediaConnChangeReason reason, const char* description)
```

【功能说明】

媒体服务器连接状态变更通知。

【回调参数】

- conn：连接对象。
- state：与媒体服务器连接状态，具体请参见**HRTCMediaConnStateTypes**。
- reason：连接状态改变原因，具体请参见**HRTCMediaConnChangeReason**。
- description：媒体连接变化原因描述。

⚠ 注意

加入房间过后，收到媒体服务的数据包时，返回Connected消息，超过6s没有收到包，则返回Failed消息。

onStartAllRemoteViewResult

```
void onStartAllRemoteViewResult(IHRTCConnection* conn, int errCode, const char* errMsg, unsigned int counts, const HRTCSetupRemoteViewResult* results)
```

【功能说明】

批量选看结果回调。

【回调参数】

- conn：连接对象。
- errCode：错误码。
- errMsg：错误信息。
- counts：results数组大小。
- results：批量选看结果，具体请参见**HRTCSetupRemoteViewResult**。

onStatsNotify

```
virtual void onStatsNotify(HRTCOnStats *rtcStats)
```

【功能说明】

当前会话统计回调。

【回调参数】

rtcStats：当前会话统计，具体请参见**HRTCOnStats**。

onVideoResolutionChangedNotify

```
void onVideoResolutionChangedNotify(IHRTCConnection* conn, const char* userId, int width, int height)
```

【功能说明】

远端视频分辨率大小改变，触发此回调。

【回调参数】

- conn：连接对象。
- userId：用户ID。
- width：视频分辨率改变后的宽。
- height：视频分辨率改变后的高。

onRemoteAudioStatsNotify

```
void onRemoteAudioStatsNotify(IHRTCConnection* conn, const HRTCRemoteAudioStats *remoteStats,
unsigned int remoteStatsCount)
```

【功能说明】

远端音频流详情，2s触发一次回调。

【回调参数】

- conn：连接对象。
- remoteStats：远端音频收流详情，具体请参见**HRTCRemoteAudioStats**。
- remoteStatsCount：remoteStats数组长度。

onRemoteVideoStatsNotify

```
void onRemoteVideoStatsNotify(IHRTCConnection* conn, const HRTCRemoteVideoStats *remoteStats,
unsigned int remoteStatsCount)
```

【功能说明】

远端视频流详情，2s触发一次回调。

【回调参数】

- conn：连接对象。
- remoteStats：远端视频收流统计，具体请参见**HRTCRemoteVideoStats**。
- remoteStatsCount：remoteStats数组长度。

onError

```
virtual void onError(IHRTCConnection* conn, int error, const char* msg)
```

【功能说明】

发生错误，触发此回调。返回客户端错误码或者服务端错误码。

【回调参数】

- conn：连接对象。
- error：错误码，具体请参见**HRTCErrorCode**。
- msg：错误描述。

onConnectionChangedNotify

```
virtual void onConnectionChangedNotify(IHRTCConnection* conn, HRTCConnStateType state,
HRTCConnChangeReason reason, const char* description)
```

【功能说明】

连接状态改变回调。

【回调参数】

- conn：连接对象。
- state：连接状态类型，具体请参见**HRTCConnStateTypes**。
- reason：连接状态改变原因，具体请参见**HRTCConnChangeReason**。
- description：连接状态改变描述。

onAuthorizationExpired

```
virtual void onAuthorizationExpired(IHRTCConnection* conn);
```

【功能说明】

鉴权签名过期回调，需要app调用**renewAuthorization**更新签名。

【回调参数】

conn：连接对象。

onJoinRoomSuccess

```
virtual void onJoinRoomSuccess(IHRTCConnection* conn, const char* userId)
```

【功能说明】

成功加入房间，触发此回调。

【回调参数】

- conn：连接对象。
- userId：新加入房间的用户ID。

onJoinRoomFailure

```
virtual void onJoinRoomFailure(IHRTCConnection* conn, int error, const char* msg)
```

【功能说明】

加入房间失败，触发此回调。

【回调参数】

- conn：连接对象。
- error：错误码，具体请参见**HRTCErrorCode**。
- msg：错误描述。

onRejoinRoomSuccess

```
virtual void onRejoinRoomSuccess(IHRTCConnection* conn, const char* userId)
```

【功能说明】

重新加入房间回调。例如，网络异常后重连成功加入房间触发。

【回调参数】

- conn：连接对象。
- userId：用户ID。

onLeaveRoom

```
virtual void onLeaveRoom(IHRTCConnection* conn, HRTCLeaveReason reason, const HRTCStatsInfo &statsInfo)
```

【功能说明】

离开房间，触发此回调。

【回调参数】

- conn：连接对象。
- reason：离开的房间原因，具体请参见**HRTCLeaveReason**。
- statsInfo：卡顿统计信息，具体请参见**HRTCStatsInfo**。

onUserRoleChangedNotify

```
virtual void onUserRoleChangedNotify(IHRTCConnection* conn, HRTCRoleType oldRole, HRTCRoleType newRole)
```

【功能说明】

用户角色切换成功，触发此回调。

【回调参数】

- conn：连接对象。
- oldRole：切换前的角色。具体请参见**HRTCRoleType**。
- newRole：切换成功后的角色。具体请参见**HRTCRoleType**。

onRemoteUserOnline

```
virtual void onRemoteUserOnline(IHRTCConnection* conn, const char* userId, const char* userName)
```

【功能说明】

远端joiner用户加入当前房间，触发此回调。该回调提示有远端joiner用户加入了房间，并返回新加入用户的ID。

【回调参数】

- conn：连接对象。
- userId：远端用户ID。
- userName：远端用户昵称。

onRemoteUserOffline

```
virtual void onRemoteUserOffline(IHRTCConnection* conn, const char* userId, int reason)
```

【功能说明】

远端joiner用户离开当前房间，触发此回调。

本端用户离开当前房间，会回调当前房间所有用户offline。

【回调参数】

- conn：连接对象。
- userId：离开房间的远端用户ID。
- reason：远端用户离线原因，预留参数。

onRemoteUserNameChangedNotify

```
virtual void onRemoteUserNameChangedNotify(IHRTCConnection* conn, const char* userId, const char*
userName)
```

【功能说明】

远端用户昵称变化，触发此回调。

【回调参数】

- conn：连接对象。
- userId：用户ID。
- userName：变更后的昵称。

onUserNameChangedNotify

```
virtual void onUserNameChangedNotify(IHRTCConnection* conn, const char* oldUserName, const char*
newUserName)
```

【功能说明】

本端用户昵称变化，触发此回调。

【回调参数】

- conn：连接对象。
- oldUserName：变更前的昵称。
- newUserName：变更后的昵称。

onRemoteAudioStateChangedNotify

```
virtual void onRemoteAudioStateChangedNotify(IHRTCConnection* conn, const char* userId,
HRTCRemoteAudioStreamState state, HRTCRemoteAudioStreamStateReason reason)
```

【功能说明】

远端音频流状态变化回调。

【回调参数】

- conn：连接对象。
- userId：远端用户ID。
- state：远端音频流状态，具体请参见**HRTCRemoteAudioStreamState**。
- reason：远端音频流状态变化原因，具体请参见**HRTCRemoteAudioStreamStateReason**。

onRemoteVideoStateChangedNotify

```
virtual void onRemoteVideoStateChangedNotify(IHRTCConnection* conn, const char* userId,
HRTCRemoteVideoStreamState state, HRTCRemoteVideoStreamStateReason reason)
```

【功能说明】

远端视频流状态变化回调。

【回调参数】

- conn：连接对象。
- userId：远端用户ID。
- state：远端视频流状态，具体请参见**HRTCRemoteVideoStreamState**。
- reason：远端视频流状态变化原因，具体请参见**HRTCRemoteVideoStreamStateReason**。

onUserVolumeStatsNotify

```
virtual void onUserVolumeStatsNotify(IHRTCConnection* conn, const HRTCVolumeInfo* userVolumes,
unsigned int userVolumesCount, unsigned int totalVolume)
```

【功能说明】

用户音量状态回调。通过**enableUserVolumeNotify**开启并设置回调周期，定时上报。

【回调参数】

- conn：连接对象。
- userVolumes：HRTCVolumeInfo，userId，volume。
- userVolumesCount：音量上报用户数组的大小。
- totalVolume：总音量。

onUserAuxiliaryStreamAvailable

```
virtual void onUserAuxiliaryStreamAvailable(IHRTCConnection* conn, const char* userId, bool available)
```

【功能说明】

远端开启，停止辅流后，触发此回调。

【回调参数】

- conn：连接对象。
- userId：远端用户ID。
- available：true表示远端开启推辅流，false表示远端停止辅流。

onFirstRemoteVideoDecoded

```
virtual void onFirstRemoteVideoDecoded(IHRTCConnection* conn, const char* userId, int width, int height)
```

【功能说明】

远端用户视频流第一帧解码成功，触发此回调。

【回调参数】

- conn：连接对象。
- userId：用户ID。
- width：视频宽。
- height：视频高。

onFirstRemoteAuxiliaryStreamDecoded

```
virtual void onFirstRemoteAuxiliaryStreamDecoded(IHRTCConnection* conn, const char* userId, int width, int height)
```

【功能说明】

远端用户辅流第一帧解码成功，触发此回调。

【回调参数】

- conn：连接对象。
- userId：用户ID。
- width：视频宽。
- height：视频高。

onAudioStatsNotify

```
virtual void onAudioStatsNotify(IHRTCConnection* conn, HRTCLocalAudioStats *localStats, unsigned int localStatsCount, HRTCRemoteAudioStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

音频流详情，2s触发一次回调。

【回调参数】

- conn：连接对象。
- localStats：本地音频发流统计，具体请参见**HRTCLocalAudioStats**。
- localStatsCount：localStats数组长度。
- remoteStats：远端音频收流详情，具体请参见**HRTCRemoteAudioStats**。
- remoteStatsCount：remoteStats数组长度。

⚠ 注意

- 当无本地音频时，localStatsCount为0，localStats为空指针。
- 当无远端音频时，remoteStatsCount为0，remoteStats为空指针。

onVideoStatsNotify

```
virtual void onVideoStatsNotify(IHRTCConnection* conn, HRTCLocalVideoStats *localStats, unsigned int localStatsCount, HRTCRemoteVideoStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

视频流详情，2s触发一次回调。

【回调参数】

- conn：连接对象。
- localStats：本地视频发流统计，具体请参见**HRTCLocalVideoStats**。
- localStatsCount：localStats数组长度。
- remoteStats：远端视频收流统计，具体请参见**HRTCRemoteVideoStats**。
- remoteStatsCount：remoteStats数组长度。

⚠ 注意

- 当无本地视频时，localStatsCount为0，localStats为空指针。
- 当无远端视频时，remoteStatsCount为0，remoteStats为空指针。

onAuxiliaryStreamStatsNotify

```
virtual void onAuxiliaryStreamStatsNotify(IHRTCConnection* conn, HRTCLocalVideoStats *localStats, unsigned int localStatsCount, HRTCRemoteVideoStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

辅流详情，2s触发一次回调。

【回调参数】

- conn：连接对象。
- localStats：本地辅流发流详情，具体请参见**HRTCLocalVideoStats**。
- localStatsCount：本地辅流发流数量。
- remoteStats：远端辅流收流详情，具体请参见**HRTCRemoteVideoStats**。
- remoteStatsCount：远端辅流收流数量。

onNetworkQualityNotify

```
virtual void onNetworkQualityNotify(IHRTCConnection* conn, HRTCQualityInfo* localQuality, unsigned int localQualityCount, HRTCQualityInfo* remoteQuality, unsigned int remoteQualityCount)
```

【功能说明】

房间内客户端基于流级别的网络质量实时上报，默认开启，每2s上报一次，有数据流时才会回调，音频流、视频流分开回调。

【回调参数】

- conn：连接对象。
- localQuality：本地上行网络质量，该参数暂时不使用。
- localQualityCount：正在上报的网络质量数量，该参数暂时不使用。

- remoteQualityCount：正在上报的流的数量，集合的大小。

onRoomStreamStatusNotify

```
virtual void onRoomStreamStatusNotify(IHRTCConnection* conn, int audienceState)
```

【功能说明】

房间流状态通知，业务调用云侧暂停/恢复接口后，端侧收到该通知。

【回调参数】

- conn：连接对象。
- audienceState: 0表示暂停 ，1表示恢复

onDestroyConnection

```
virtual void onDestroyConnection(IHRTCConnection* conn)
```

【功能说明】

调用release释放连接后，触发该回调，这是IHRTCConnection对象最后一个回调，需要在该回调触发后再去清理IHRTCConnectionEventHandler资源。

【回调参数】

conn：连接对象。

onError

```
virtual void onError(int error, const char* msg)
```

【功能说明】

发生错误，触发此回调。返回**客户端错误码**或者**服务端错误码**。

【回调参数】

- error：错误码，具体请参见**HRTCErrorCode**。
- msg：错误描述。

onJoinRoomSuccess

```
virtual void onJoinRoomSuccess(const char* roomId, const char* userId)
```

【功能说明】

成功加入房间，触发此回调。

【回调参数】

- roomId：新加入的房间ID。
- userId：新加入房间的用户ID。

onJoinRoomFailure

```
virtual void onJoinRoomFailure(int error, const char* msg)
```

【功能说明】

加入房间失败，触发此回调。

【回调参数】

- error：错误码，具体请参见**HRTCErrorCode**。
- msg：错误描述。

onLeaveRoom

```
virtual void onLeaveRoom(HRTCLeaveReason reason, const HRTCStatsInfo &statsInfo)
```

【功能说明】

离开房间，触发此回调。

【回调参数】

- reason：离开的房间原因，具体请参见**HRTCLeaveReason**。
- statsInfo：卡顿统计信息，具体请参见**HRTCStatsInfo**。

⚠ 注意

APP调用**leaveRoom**接口时，会返回**HRTC_LEAVE_REASON_USER_LEAVE_ROOM**，可以通过以下任一方式回退到登录界面。

- APP在调用**leaveRoom**接口时退到登录界面，或者在收到onLeaveRoom回调，且回调消息不等于**HRTC_LEAVE_REASON_USER_LEAVE_ROOM**时（防止重复操作）退到登录界面。
- APP只在收到onLeaveRoom消息时退到登录界面。

onRemoteUserOnline

```
virtual void onRemoteUserOnline(const char* roomId, const char* userId, const char* userName)
```

【功能说明】

远端joiner用户加入当前房间，触发此回调。该回调提示有远端joiner用户加入了房间，并返回新加入用户的ID。

【回调参数】

- roomId：房间ID。
- userId：远端用户ID。
- userName：远端用户昵称。

onRemoteUserOffline

```
virtual void onRemoteUserOffline(const char* roomId, const char* userId, int reason)
```

【功能说明】

远端joiner用户离开当前房间，触发此回调。

本端用户离开当前房间，会回调当前房间所有用户offline。

【回调参数】

- roomId：当前房间的房间ID。
- userId：离开房间的远端用户ID。
- reason：远端用户离线原因，预留参数。

onRemoteUserNameChangedNotify

```
virtual void onRemoteUserNameChangedNotify(const char* roomId, const char* userId, const char* userName)
```

【功能说明】

远端用户昵称变化，触发此回调。

【回调参数】

- roomId：房间ID。
- userId：用户ID。
- userName：变更后的昵称。

onUserNameChangedNotify

```
virtual void onUserNameChangedNotify(const char* oldUserName, const char* newUserName)
```

【功能说明】

本端用户昵称变化，触发此回调。

【回调参数】

- oldUserName：变更前的昵称。
- newUserName：变更后的昵称。

onFirstRemoteAuxiliaryStreamDecoded

```
virtual void onFirstRemoteAuxiliaryStreamDecoded(const char* roomId, const char* userId, int width, int height)
```

【功能说明】

接收到第一帧远端辅流并解码成功，触发此回调。

【回调参数】

- roomId：视频流对应的房间ID。
- userId：视频流对应的用户ID。
- width：视频流宽。
- height：视频流高。

onFirstRemoteVideoDecoded

```
virtual void onFirstRemoteVideoDecoded(const char* roomId, const char* userId, int width, int height)
```

【功能说明】

接收到第一帧远端视频流并解码成功，触发此回调。

【回调参数】

- roomId：视频流对应的房间ID。
- userId：视频流对应的用户ID。
- width：视频流宽。
- height：视频流高。

onConnectionChangedNotify

```
virtual void onConnectionChangedNotify(HRTCConnStateTypes connType, HRTCConnChangeReason reason, const char* description)
```

【功能说明】

网络连接状态发生变化，触发此回调。

【回调参数】

- connType：网络连接状态。具体请参见**HRTCConnStateTypes**。
- reason：网络连接状态发生变化原因。具体请参见**HRTCConnChangeReason**。
- description：错误原因描述。

onDeviceStateChangedNotify

```
virtual void onDeviceStateChangedNotify(const char* deviceId, HRTCDeviceType deviceType, HRTCDeviceState deviceState)
```

【功能说明】

设备状态发生变化，触发此回调。

【回调参数】

- deviceId：系统设备标识，如系统音频播放设备标识可通过**getPlaybackDevices**获取。
- deviceType：系统设备类型，具体请参见**HRTCDeviceType**。
- deviceState：系统设备状态，具体请参见**HRTCDeviceState**。

⚠ 注意

通话前插拔设备会上报变化。

onDeviceVolumeChangedNotify

```
virtual void onDeviceVolumeChangedNotify(HRTCDeviceType deviceType, unsigned int volume, unsigned int muted)
```

【功能说明】

音频设备音量发生变化，触发此回调。

【回调参数】

- deviceType：系统设备类型，具体请参见**HRTCDeviceType**。
- volume：音量
- muted：true表示设备静音，false表示设备未静音。

⚠ 注意

通话前调整音频设备音量和静音会上报变化。

onLogUploadResult

```
virtual void onLogUploadResult(int result)
```

【功能说明】

日志上传结果回调。

【回调参数】

result：日志上传结果。

onLogUploadProgress

```
virtual void onLogUploadProgress(int progress)
```

【功能说明】

日志上传进度回调。

【回调参数】

progress：日志上传进度。取值范围为[0,100]。

onUserRoleChangedNotify

```
virtual void onUserRoleChangedNotify(const char* roomId, HRTCRoleType oldRole, HRTCRoleType newRole)
```

【功能说明】

用户角色切换成功，触发此回调。

【回调参数】

- roomId：发生角色切换的房间号。
- oldRole：切换前的角色。具体请参见**HRTCRoleType**。
- newRole：切换成功后的角色。具体请参见**HRTCRoleType**。

onScreenShareStarted

```
virtual void onScreenShareStarted()
```

【功能说明】

屏幕流共享开启，触发此回调。

onScreenShareStopped

```
virtual void onScreenShareStopped(int reason)
```

【功能说明】

屏幕流共享关闭，触发此回调。

【回调参数】

reason：屏幕共享关闭原因。

onUserAuxiliaryStreamAvailable

```
virtual void onUserAuxiliaryStreamAvailable(const char* roomId, const char* userId, bool available)
```

【功能说明】

远端开启，停止辅流后，触发此回调。

【回调参数】

- roomId：房间ID。
- userId：远端用户ID。
- available：true表示远端开启辅流，false表示远端停止辅流。

onVideoStatsNotify

```
virtual void onVideoStatsNotify(HRTCLocalVideoStats *localStats, unsigned int localStatsCount,
HRTCRemoteVideoStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

视频流详情，2s触发一次回调。

【回调参数】

- localStats：本地视频发流统计，具体请参见**HRTCLocalVideoStats**。
- localStatsCount：localStats数组长度。
- remoteStats：远端视频收流统计，具体请参见**HRTCRemoteVideoStats**。
- remoteStatsCount：remoteStats数组长度。

⚠ 注意

- 当无本地视频时，localStatsCount为0，localStats为空指针，需要先判断不为空再使用，否则可能引发空指针错误。
- 当无远端视频时，remoteStatsCount为0，remoteStats为空指针，需要先判断不为空再使用，否则可能引发空指针错误。

onAudioStatsNotify

```
virtual void onAudioStatsNotify(HRTCLocalAudioStats *localStats, unsigned int localStatsCount,
HRTCRemoteAudioStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

音频流详情，2s触发一次回调。

【回调参数】

- localStats：本地音频发流统计，具体请参见**HRTCLocalAudioStats**。
- localStatsCount：localStats数组长度。
- remoteStats：远端音频收流统计，具体请参见**HRTCRemoteAudioStats**。
- remoteStatsCount：remoteStats数组长度。

⚠ 注意

- 当无本地音频时，localStatsCount为0，localStats为空指针，需要先判断不为空再使用，否则可能引发空指针错误。
- 当无远端音频时，remoteStatsCount为0，remoteStats为空指针，需要先判断不为空再使用，否则可能引发空指针错误。

onAuxiliaryStreamStatsNotify

```
virtual void onAuxiliaryStreamStatsNotify(HRTCLocalVideoStats *localStats, unsigned int localStatsCount, HRTCRemoteVideoStats *remoteStats, unsigned int remoteStatsCount)
```

【功能说明】

辅流详情，2s触发一次回调。

【回调参数】

- localStats：本地辅流发流统计，具体请参见**HRTCLocalVideoStats**。
- localStatsCount：localStats数组长度。
- remoteStats：远端辅流收流统计，具体请参见**HRTCRemoteVideoStats**。
- remoteStatsCount：remoteStats数组长度。

⚠ 注意

- 当无本地辅流时，localStatsCount为0，localStats为空指针，需要先判断不为空再使用，否则可能引发空指针错误。
- 当无远端辅流时，remoteStatsCount为0，remoteStats为空指针，需要先判断不为空再使用，否则可能引发空指针错误。

onAuthorizationExpired

```
virtual void onAuthorizationExpired();
```

【功能说明】

鉴权签名过期回调，需要app调用更新签名。

onRemoteAudioStateChangedNotify

```
virtual void onRemoteAudioStateChangedNotify(const char* userId, HRTCRemoteAudioStreamState state, HRTCRemoteAudioStreamStateReason reason)
```

【功能说明】

远端音频流状态变化回调。

【回调参数】

- userId：远端用户ID。
- state：远端音频流状态，具体请参见**HRTCRemoteAudioStreamState**。
- reason：远端音频流状态变化原因，具体请参见**HRTCRemoteAudioStreamStateReason**。

onRemoteVideoStateChangedNotify

```
virtual void onRemoteVideoStateChangedNotify(const char* userId, HRTCRemoteVideoStreamState state, HRTCRemoteVideoStreamStateReason reason)
```

【功能说明】

远端视频流状态变化回调。

【回调参数】

- userId：远端用户ID。
- state：远端视频流状态，具体请参见**HRTCRemoteVideoStreamState**。
- reason：远端视频流状态变化原因，具体请参见**HRTCRemoteVideoStreamStateReason**。

onRenderAuxiliaryExternalVideoFrame

```
virtual void onRenderAuxiliaryExternalVideoFrame(const char* roomId, HRTCMediaDirection direction, const char* userId, HRTCVideoFrame& videoFrame)
```

【功能说明】

辅流自渲染回调。需要调用setAuxiliaryExternalVideoFrameOutput接口开启辅流自渲染，从而触发该回调。

【回调参数】

- roomId：房间ID。
- direction：数据源，本地数据，远端数据，具体请参见**HRTCMediaDirection**。
- userId：用户ID。
- videoFrame：辅流详情，具体请参见**HRTCVideoFrame**。

onRejoinRoomSuccess

```
virtual void onRejoinRoomSuccess(const char* roomId, const char* userId)
```

【功能说明】

重新加入房间回调。例如网络异常后重连成功加入房间触发。

【回调参数】

- roomId：房间ID。
- userId：用户ID。

onNetworkTestQualitysection1015722617613

```
virtual void onNetworkTestQuality(HRTCNetworkQualityLevel level)
```

【功能说明】

加房间前网络探测回调。

【回调参数】

level：网络质量，具体请参见**HRTCNetworkQualityLevel**。

onNetworkTestResult

```
virtual void onNetworkTestResult(HRTCNetworkTestResult& networkTestResult)
```

【功能说明】

加房间前网络探测结果回调。

【回调参数】

networkTestResult：主要包括测试成功与否、上行和下行的网络带宽、丢包、延时和抖动，具体请参见**HRTCNetworkTestResult**。

onUserVolumeStatsNotify

```
virtual void onUserVolumeStatsNotify(const HRTCVolumeInfo* userVolumes, unsigned int
userVolumesCount, unsigned int totalVolume)
```

【功能说明】

用户音量状态回调。通过**enableUserVolumeNotify**开启并设置回调周期，定时上报。

【回调参数】

- userVolumes：用户信息，具体请参见**HRTCVolumeInfo**。
- userVolumesCount：上报的用户人数，包含本地用户。
- totalVolume：总音量。

onTopActiveSpeaker

```
virtual void onTopActiveSpeaker(const char* userId, bool noStream)
```

【功能说明】

声控画面的用户ID变化时，触发此回调。该回调主要用于0号会场场景。

【回调参数】

userId：返回当前声控画面的用户ID。

noStream：该用户是否有视频流。

⚠ 注意

0号会场模式下，SDK会持续监测（根据一定时间内用户音量大小）当前最活跃的用户，如果最活跃用户发生变化，则触发此回调并上报当前最活跃的用户userId。

onLocalAudioStateChangedNotify

```
virtual void onLocalAudioStateChangedNotify(HRTCLocalAudioStreamState state,
HRTCLocalAudioStreamStateReason reason)
```

【功能说明】

本地音频状态改变，触发此回调。

【回调参数】

- state：本地音频状态，具体请参见**HRTCLocalAudioStreamState**。
- reason：本地音频状态改变的原因，具体请参见**HRTCLocalAudioStreamStateReason**。

onLocalVideoStateChangedNotify

```
virtual void onLocalVideoStateChangedNotify(HRTCLocalVideoStreamState state,
HRTCLocalVideoStreamStateReason reason)
```

【功能说明】

本地视频状态改变，触发此回调。

【回调参数】

- state：本地视频状态，具体请参见**HRTCLocalVideoStreamState**。
- reason：本地视频状态改变原因，具体请参见**HRTCLocalVideoStreamStateReason**。

onMediaConnectStateChangedNotify

```
virtual void onMediaConnectStateChangedNotify(HRTCMediaConnStateTypes state,
HRTCMediaConnChangeReason reason, const char* description)
```

【功能说明】

媒体服务器连接状态变更通知。

【回调参数】

- state：与媒体服务器连接状态，具体请参见**HRTCMediaConnStateTypes**。
- reason：连接状态变化的原因，具体请参见**HRTCMediaConnChangeReason**。
- description：连接状态变化原因描述。

⚠ 注意

加入房间过后，收到媒体服务的数据包时，返回Connected消息，超过6s没有收到包，则返回Failed消息。

onStartAllRemoteViewResult

```
virtual void onStartAllRemoteViewResult(int errCode, const char* errMsg, unsigned int counts, const
HRTCSetupRemoteViewResult* results)
```

【功能说明】

批量选看结果回调。

【回调参数】

- errCode：错误码。
- errMsg：错误信息。
- counts：results数组大小。
- results：批量选看结果，具体请参见**HRTCSetupRemoteViewResult**。

onStatsNotify

```
virtual void onStatsNotify(HRTCOnStats *rtcStats)
```

【功能说明】

当前会话统计回调。

【回调参数】

rtcStats：当前会话统计，具体请参见**HRTCOnStats**。

onLocalVideoStatsNotify

```
virtual void onLocalVideoStatsNotify(const HRTCLocalVideoStats *localStats, unsigned int localStatsCount)
```

【功能说明】

本地视频流详情，2s触发一次回调。

【回调参数】

- localStats：本地视频收流统计，具体请参见**HRTCLocalVideoStats**。
- localStatsCount：localStats数组长度。

onRemoteVideoStatsNotify

```
virtual void onRemoteVideoStatsNotify(const HRTCRemoteVideoStats *remoteStats, unsigned int
remoteStatsCount)
```

【功能说明】

远端视频流详情，2s触发一次回调。

【回调参数】

- remoteStats：远端视频收流统计，具体请参见**HRTCRemoteVideoStats**。
- remoteStatsCount：remoteStats数组长度。

onLocalAudioStatsNotify

```
virtual void onLocalAudioStatsNotify(const HRTCLocalAudioStats *localStats, unsigned int localStatsCount)
```

【功能说明】

本地音频流详情，2s触发一次回调。

【回调参数】

- localStats：本地音频收流统计，具体请参见**HRTCLocalAudioStats**。
- localStatsCount：localStats数组长度。

onRemoteAudioStatsNotify

```
virtual void onRemoteAudioStatsNotify(const HRTCRemoteAudioStats *remoteStats, unsigned int
remoteStatsCount)
```

【功能说明】

远端音频流详情，2s触发一次回调。

【回调参数】

- remoteStats：远端音频收流统计，具体请参见**HRTCRemoteAudioStats**。
- remoteStatsCount：remoteStats数组长度。

onVideoResolutionChangedNotify

```
virtual void onVideoResolutionChangedNotify(const char* userId, int width, int height)
```

【功能说明】

远端视频分辨率大小改变，触发此回调。

【回调参数】

- userId：用户ID。
- width：视频分辨率改变后的宽。
- height：视频分辨率改变后的高。

onAudioClipFinished

```
virtual void onAudioClipFinished(int soundId)
```

【功能说明】

音效文件播放结束，触发此回调。

【回调参数】

soundId：音效ID，取值>=0。

onAudioFileFinished

```
virtual void onAudioFileFinished()
```

【功能说明】

音频文件播放结束，触发此回调。

【回调参数】

无

onAudioMixStateChangedNotify

```
virtual void onAudioMixStateChangedNotify(HRTCAudioFileState state, HRTCAudioFileReason reason,
unsigned long long value)
```

【功能说明】

混音音频文件播放状态改变，触发此回调。

【回调参数】

- state：音频播放状态，具体请参见**HRTCAudioFileState**。
- reason：音频播放状态改变原因，具体请参见**HRTCAudioFileReason**。
- value：state为HWRtcAudioFileOpenCompleted表示音频文件的时长，单位为ms；state为HWRtcAudioFilePositionUpdate表示当前播放的位置，单位为ms。其他情况下，value值无意义。

onSeiSendMsgSuccess

```
void onSeiSendMsgSuccess(const char* message);
```

【功能说明】

音频SEI信息发送成功回调。

回调参数

message：发送SEI信息的内容。

onSeiRecvMsg

```
void onSeiRecvMsg(const char* userId, const char* message);
```

【功能说明】

接收音频SEI信息回调。

回调参数

- userId：用户ID。
- message：接收SEI信息的内容。

onStartPublishStream

```
void onStartPublishStream(int code, const char* taskId);
```

【功能说明】

开始旁路（RTMP）推流回调。

回调参数

- code：错误码，成功为0，失败参考错误码**HRTCErrorCode**。
- taskId：任务Id。

onUpdateTransCoding

```
void onUpdateTransCoding(int code, const char* taskId);
```

【功能说明】

更新旁路（RTMP）推流消息。

回调参数

- code：错误码，成功为0，失败参考错误码**HRTCErrorCode**。
- taskId：任务Id。

onStopPublishStream

```
void onStopPublishStream(int code, const char* taskId);
```

【功能说明】

停止旁路（RTMP）推流消息。

回调参数

- code：错误码，成功为0，失败参考错误码**HRTCErrorCode**。
- taskId：任务Id。

onStreamPublishStateChange

```
void onStreamPublishStateChange(int code, const char* taskId, const HRTCUrlStatusList * urlStatu);
```

【功能说明】

RTMP推流状态回调。

回调参数

- code：错误码，成功为0，失败参考错误码**HRTCErrorCode**。
- taskId：任务Id。
- urlStatu：推流的url状态，具体请参见**HRTCUrlStatusList**。

onMultiRoomMediaRelayStateChanged

```
void onMultiRoomMediaRelayStateChanged(const char *roomId, HRTCMultiRoomMediaRelayState state,
HRTCMultiRoomMediaRelayStateCode code);
```

【功能说明】

跨房状态回调。

【回调参数】

- roomId：跨房房间号。
- state：状态类型，具体请参见**HRTCMultiRoomMediaRelayState**。
- code：状态的具体原因，具体请参见**HRTCMultiRoomMediaRelayStateCode**。

onRemoteMicrophoneStateChanged

```
void onRemoteMicrophoneStateChanged(const char* userId, HRTCRemoteMicState state);
```

【功能说明】

远端麦克风设备状态变更通知。

【回调参数】

- userId：远端用户userId。
- state：麦克风设备状态，具体请参见**HRTCRemoteMicState**。

7.4.5 音频设备管理

本章节介绍了全平台C++ SDK的IHRTCAudioDeviceManager接口详情。

表 7-17 IHRTCAudioDeviceManager 接口

接口	描述
getPlaybackDevices	获取系统音频播放设备列表
getRecordDevices	获取系统音频录制设备列表

接口	描述
setPlaybackDevice	指定音频播放设备
setRecordDevice	指定音频录制设备
setPlaybackDeviceVolume	设置音频播放设备音量
setRecordDeviceVolume	设置音频录制设备音量
getPlaybackDeviceVolume	获取音频播放设备音量
getRecordDeviceVolume	获取音频录制设备音量
setPlaybackDeviceMuteState	设置音频播放设备是否静音
setRecordDeviceMuteState	设置音频录制设备是否静音
getPlaybackDeviceMuteState	获取音频播放设备静音状态
getRecordDeviceMuteState	获取音频录制设备静音状态
getCurrentPlaybackDevice	获取当前音频播放设备
getCurrentRecordDevice	获取当前音频录制设备
recordingDeviceTest	音频采集设备测试（只支持Windows和macOS）
finishRecordingDeviceTest	结束音频采集设备测试（只支持Windows和macOS）
playbackDeviceTest	音频播放设备测试（只支持Windows和macOS）
finishPlaybackDeviceTest	结束音频播放设备测试（只支持Windows和macOS）
echoTest	音频设备回路测试（只支持Windows和macOS）
finishEchoTest	结束音频设备回路测试（只支持Windows和macOS）

getPlaybackDevices

```
virtual int getPlaybackDevices(HRTCDeviceInfo *deviceInfo, unsigned int *counts)
```

【功能说明】

获取系统音频播放设备列表。

【请求参数】

- deviceInfo：输出参数，获取系统设备详情，包括设备标识和设备名称。具体请参见HRTCDeviceInfo。
- counts：输入输出参数，获取deviceInfo对象数量，输入值为deviceInfo的数量，将会返回小于等于counts数量的设备信息。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

getRecordDevices

```
virtual int getRecordDevices(HRTCDeviceInfo *deviceInfo, unsigned int *counts)
```

【功能说明】

获取系统音频录制设备列表。

【请求参数】

- deviceInfo：输出参数，获取系统设备详情，包括设备标识和设备名称。具体请参见**HRTCDeviceInfo**。
- counts：输入输出参数，获取deviceInfo对象数量，输入值为deviceInfo的数量，将会返回小于等于counts数量的设备信息。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setPlaybackDevice

```
virtual int setPlaybackDevice(const char deviceId[HRTC_MAX_DEVICE_ID_LEN + 1])
```

【功能说明】

指定音频播放设备。

【请求参数】

deviceId：设备标识。可通过**getPlaybackDevices**获取。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 通话前没有选择任何设备，通话中使用默认设备。
- 通话前已选择指定设备，通话中直接生效。
- 通话中有多个音频设备，拔出设备会自动切换。
- 通话中无音频设备，新插入设备，需要重新调用该接口指定设备。

setRecordDevice

```
virtual int setRecordDevice(const char deviceId[HRTC_MAX_DEVICE_ID_LEN+ 1])
```

【功能说明】

指定音频录制设备。

【请求参数】

deviceId：设备标识。可通过**getRecordDevices**获取。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 通话前没有选择任何设备，通话中使用默认设备。
- 通话前已选择指定设备，通话中直接生效。
- 通话中有多个音频设备，拔出设备会自动切换。
- 通话中无音频设备，新插入设备，需要重新调用该接口指定设备。

setPlaybackDeviceVolume

```
virtual int setPlaybackDeviceVolume(unsigned int volume)
```

【功能说明】

设置音频播放设备音量。

【请求参数】

volume：播放设备音量。取值范围为[0,100]。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setRecordDeviceVolume

```
virtual int setRecordDeviceVolume(unsigned int volume)
```

【功能说明】

设置音频录制设备音量。

【请求参数】

volume：录音设备音量。取值范围为[0,100]。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

getPlaybackDeviceVolume

```
virtual int getPlaybackDeviceVolume(unsigned int *volume)
```

【功能说明】

获取音频播放设备音量。

【请求参数】

volume：播放设备音量，取值范围为[0,100]。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

getRecordDeviceVolume

```
virtual int getRecordDeviceVolume(unsigned int *volume)
```

【功能说明】

获取音频录制设备音量。

【请求参数】

volume：输出参数，录音设备音量。取值范围为[0,100]。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setPlaybackDeviceMuteState

```
virtual int setPlaybackDeviceMuteState(bool mute)
```

【功能说明】

设置音频播放设备是否静音。

【请求参数】

mute：true表示设备设为静音，false表示设备设为不静音。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

setRecordDeviceMuteState

```
virtual int setRecordDeviceMuteState(bool mute)
```

【功能说明】

设置音频录制设备是否静音。

【请求参数】

mute：true表示设备设为静音，false表示设备设为非静音。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

getPlaybackDeviceMuteState

```
virtual int getPlaybackDeviceMuteState(bool *mute)
```

【功能说明】

获取音频播放设备静音状态。

【请求参数】

mute：true表示已静音状态，false表示非静音状态。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

getRecordDeviceMuteState

```
virtual int getRecordDeviceMuteState(bool *mute)
```

【功能说明】

获取音频录制设备静音状态。

【请求参数】

mute：true表示已静音状态，false表示非静音状态。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

getCurrentPlaybackDevice

```
virtual int getCurrentPlaybackDevice(char deviceId[HRTC_MAX_DEVICE_ID_LEN + 1])
```

【功能说明】

获取当前音频播放设备。

【请求参数】

deviceId：输出参数，当前播放设备标识。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

getCurrentRecordDevice

```
virtual int getCurrentRecordDevice(char deviceId[HRTC_MAX_DEVICE_ID_LEN+ 1])
```

【功能说明】

获取当前音频录制设备。

【请求参数】

deviceId：输出参数，当前录制设备标识。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

recordingDeviceTest

```
virtual int recordingDeviceTest(int intervalInMilliseconds)
```

【功能说明】

音频采集设备测试。

【请求参数】

intervalInMilliseconds：音量回调周期，单位毫秒，范围100到10000。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

只支持Windows和macOS。

finishRecordingDeviceTest

```
virtual int finishRecordingDeviceTest()
```

【功能说明】

结束音频采集设备测试。

【请求参数】

无。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

只支持Windows和macOS。

playbackDeviceTest

```
virtual int playbackDeviceTest(const char *testAudioFilePath)
```

【功能说明】

音频播放设备测试。

【请求参数】

testAudioFilePath：测试音频文件，只支持WAV格式。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

只支持Windows和macOS。

finishPlaybackDeviceTest

```
virtual int finishPlaybackDeviceTest()
```

【功能说明】

结束音频播放设备测试。

【请求参数】

d无。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

只支持Windows和macOS。

echoTest

```
virtual int echoTest(int intervalInMilliseconds)
```

【功能说明】

音频设备回路测试。

【请求参数】

intervalInMilliseconds：音量回调周期，单位毫秒，范围100到10000。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

只支持Windows和macOS。

finishEchoTest

```
virtual int finishEchoTest()
```

【功能说明】

结束音频设备回路测试。

【请求参数】

无。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见HRTCErrorCode。

注意

只支持Windows和macOS。

7.4.6 视频设备管理

本章节介绍了全平台C++ SDK的IHRTCVideoDeviceManager接口详情。

表 7-18 IHRTCVideoDeviceManager 接口

接口	描述
getVideoDevices	获取系统视频设备列表
setVideoDevice	设置视频设备
getCurrentVideoDevice	获取当前视频设备
cameraDeviceTest	视频采集设备测试
finishCameraDeviceTest	结束视频采集设备测试

表 7-19 IVideoDeviceCollection 类

接口	描述
getCount	获取设备个数
getDevice	通过设备序号获取设备信息
release	释放接口

getVideoDevices

```
virtual int getVideoDevices(HRTCDeviceInfo *deviceInfo, unsigned int *counts)
```

【功能说明】

获取系统视频设备列表。

【请求参数】

- deviceInfo：输出参数，获取系统设备详情，包括设备标识和设备名称。具体请参见**HRTCDeviceInfo**。
- counts：输入输出参数，获取deviceInfo对象数量，输入值为deviceInfo的数量，将会返回小于等于counts数量的设备信息。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

getVideoDevices

```
virtual IVideoDeviceCollection * getVideoDevices()
```

【功能说明】

获取系统视频设备列表。

【返回参数】

IVideoDeviceCollection：视频设备列表，具体请参见**IVideoDeviceCollection**。

setVideoDevice

```
virtual int setVideoDevice(const char deviceId[HRTC_MAX_DEVICE_ID_LEN+ 1])
```

【功能说明】

设置视频设备。

【请求参数】

deviceId：设备标识。可通过**getVideoDevices**获取。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

⚠ 注意

- 通话前没有选择任何设备，通话中使用默认设备。
- 通话前已选择指定设备，通话中直接生效。
- 通话中所有拔插视频设备的操作，不会自动切换设备，需要应用重新调用该接口并设置对应设备才生效。

getCurrentVideoDevice

```
virtual int getCurrentVideoDevice(char deviceId[HRTC_MAX_DEVICE_ID_LEN + 1])
```

【功能说明】

获取当前视频设备。

【请求参数】

deviceId：输出参数。设备标识。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

cameraDeviceTest

```
virtual int cameraDeviceTest(view_t hwnd)
```

【功能说明】

视频采集设备测试，只支持入会前调用。

【请求参数】

hwnd：窗口句柄。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

finishCameraDeviceTest

```
virtual int finishCameraDeviceTest()
```

【功能说明】

结束视频采集设备测试，只支持入会前调用。

【请求参数】

无。

【返回参数】

- 0：方法调用成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

getCount

```
virtual int getCount()
```

【功能说明】

获取当前视频设备个数。

【返回参数】

视频设备个数。

getDevice

```
virtual HRTCDeviceInfo* getDevice(int index)
```

【功能说明】

通过视频序号获取视频设备信息。一般轮询取设备信息，index从0到count取到的个数减1。

【返回参数】

视频设备信息，具体请参见**HRTCDeviceInfo**。

release

```
virtual void release()
```

【功能说明】

释放接口。

7.4.7 共享屏幕资源管理

本章节介绍了全平台C++ SDK的IHRTCScreenShareSourceList类接口详情。

表 7-20 IHRTCScreenShareSourceList 类

接口	描述
count	获取屏幕共享窗口资源数量
get	获取屏幕共享窗口资源信息
release	资源释放函数

count

```
virtual unsigned int count()
```

【功能说明】

获取屏幕共享窗口资源数量。

【请求参数】

无

【返回参数】

unsigned int：无符号整数，对象（屏幕或窗口）数量。

get

```
virtual HRTCScreenShareSourceInfo get(unsigned int index)
```

【功能说明】

获取屏幕共享窗口资源信息。

【请求参数】

index：资源句柄。

【返回参数】

捕获的共享资源对象，具体请参见**HRTCScreenShareSourceInfo**。

release

```
virtual void release()
```

【功能说明】

资源释放函数。

7.4.8 媒体原始数据管理

7.4.8.1 注册回调（IHRTCMediaEngine）

本章节介绍了全平台C++ SDK的IHRTCMediaEngine类接口详情。

表 7-21 IHRTCMediaEngine 类

接口	描述
setVideoFrameObserver	注册原始视频媒体数据监听回调
setAudioFrameObserver	注册原始音频媒体数据监听回调

setVideoFrameObserver

```
virtual int setVideoFrameObserver(IHRTCVideoFrameObserver* observer)
virtual int setVideoFrameObserver(IHRTCConnection* conn, IHRTCConnectionVideoFrameObserver*
observer)
```

【功能说明】

注册原始视频媒体数据监听回调。

【请求参数】

- conn：连接对象。
- IHRTCVideoFrameObserver* observer：原始视频数据处理接口代理。具体请参见IHRTCVideoFrameObserver。
- IHRTCConnectionVideoFrameObserver* observer：原始视频数据处理接口代理。具体请参见IHRTCVideoFrameObserver。

【返回参数】

- 0：成功。
- > 0：方法调用失败。具体请参见HRTCErrorCode。

setAudioFrameObserver

```
virtual int setAudioFrameObserver(IHRTCAudioFrameObserver* observer )
```

【功能说明】

注册原始音频媒体数据监听回调。

【请求参数】

observer：原始音频数据处理接口代理。具体请参见IHRTCAudioFrameObserver。

【返回参数】

- 0：成功。
- > 0：方法调用失败。具体请参见**HRTCErrorCode**。

7.4.8.2 事件回调(IHRTCVideoFrameObserver)

本章节介绍了全平台C++ SDK的回调接口IHRTCVideoFrameObserver的详情。

表 7-22 事件回调说明

接口	描述
onVideoFrameCapture	原始视频回调（前处理）
onVideoFrameRender	渲染后视频回调（后处理）
requireCaptureVideoFrame	是否开启前处理
requireRenderVideoFrame	是否开启后处理

onVideoFrameCapture

```
virtual bool onVideoFrameCapture(HRTCVideoFrame& videoFrame)
```

【功能说明】

原始视频回调，从接口回调中取到原始视频数据以作前处理。

【回调参数】

videoFrame：视频数据格式，具体请参见**HRTCVideoFrame**。

【返回参数】

- true：处理结果成功。
- false：处理结果失败。

onVideoFrameRender

```
virtual bool onVideoFrameRender(const char* userId, HRTCVideoFrame& videoFrame)
```

【功能说明】

原始视频数据处理后回调

【回调参数】

- userid：用户ID
- videoFrame：视频数据格式，具体请参见**HRTCVideoFrame**。

【返回参数】

- true：处理结果成功。
- false：处理结果失败。

requireCaptureVideoFrame

```
virtual bool requireCaptureVideoFrame()
```

【功能说明】

是否需要开启前处理。

【返回参数】

- true：开启。
- false：不开启。

requireRenderVideoFrame

```
virtual bool requireRenderVideoFrame()
```

【功能说明】

是否需要开启后处理。

【返回参数】

- true：开启。
- false：不开启。

7.4.8.3 事件回调(IHRTCAudioFrameObserver)

本章节介绍了全平台C++ SDK的回调接口IHRTCAudioFrameObserver的详情。

表 7-23 事件回调说明

接口	描述
onAudioFramePlayback	音频播放回调（后处理）
onAudioFrameMixed	音频混音处理回调
onAudioFrameRecord	音频采集回调（前处理）
requireRecordAudioFrame	是否开启音频前处理
requirePlaybackAudioFrame	是否开启音频后处理
requireMixedAudioFrame	是否开启音频混音回调

onAudioFramePlayback

```
virtual bool onAudioFramePlayback(HRTCAudioFrame& audioFrame)
```

【功能说明】

需要播放的音频数据回调，从接口回调中取到音频数据以作后处理。

【回调参数】

audioFrame：音频数据格式，具体请参见**HRTCAudioFrame**。

【返回参数】

- true：处理结果成功。
- false：处理结果失败。

onAudioFrameMixed

```
virtual bool onAudioFrameMixed(HRTCAudioFrame& audioFrame)
```

【功能说明】

全部音频混音数据回调，包含上下行所有通道。

【回调参数】

audioFrame：音频数据格式 ，具体请参见**HRTCAudioFrame**。

【返回参数】

- true：处理结果成功。
- false：处理结果失败。

onAudioFrameRecord

```
virtual bool onAudioFrameRecord(HRTCAudioFrame& audioFrame)
```

【功能说明】

音频采集原始数据回调，对音频数据的修改会发送到远端。

【回调参数】

audioFrame：音频数据格式 ，具体请参见**HRTCAudioFrame**。

【返回参数】

- true：处理结果成功。
- false：处理结果失败。

requireRecordAudioFrame

```
virtual bool requireRecordAudioFrame()
```

【功能说明】

是否开启音频前处理。

【返回参数】

- true：开启。
- false：不开启。

requirePlaybackAudioFrame

```
virtual bool requirePlaybackAudioFrame()
```

【功能说明】

是否需要开启音频后处理。

【返回参数】

- true：开启。
- false：不开启。

requireMixedAudioFrame

```
virtual bool requireMixedAudioFrame()
```

【功能说明】

是否需要开启全部音频混音数据回调。

【返回参数】

- true：开启。
- false：不开启。

7.4.8.4 事件回调(IHRTCConnectionVideoFrameObserver)

本章节介绍了全平台C++ SDK的回调接口IHRTCConnectionVideoFrameObserver的详情。

表 7-24 事件回调说明

接口	描述
onVideoFrameRender	渲染后视频回调（后处理）

onVideoFrameRender

```
virtual bool onVideoFrameRender(IHRTCConnection* conn, const char* userId, HRTCVideoFrame& videoFrame)
```

【功能说明】

原始视频数据处理后回调

【回调参数】

- conn：连接对象
- userid：用户ID
- videoFrame：视频数据格式，具体请参见HRTCVideoFrame。

【返回参数】

- true：处理结果成功。
- false：处理结果失败。

7.4.8.5 事件回调（IHRTCEncDecryptFrameObserver）

表 7-25 事件回调说明

接口	描述
onMediaFrameEncrypt	加密回调

接口	描述
onMediaFrameDecrypt	解密回调

onMediaFrameEncrypt

```
virtual bool onMediaFrameEncrypt(HRTCMediaType frameType, HRTCFrameBuffer &frameBuffer)
```

【功能说明】

媒体数据的加密回调

【回调参数】

- frameType：媒体类型。
- frameBuffer：媒体数据格式，具体请参见HRTCFrameBuffer。

onMediaFrameDecrypt

```
virtual bool onMediaFrameDecrypt(HRTCMediaType frameType, HRTCFrameBuffer &frameBuffer)
```

【功能说明】

媒体数据的解密密回调

【回调参数】

- frameType：媒体类型
- frameBuffer：媒体数据格式，具体请参见HRTCFrameBuffer。

7.4.9 客户端错误码

本章节介绍了SDK的客户端错误码HRTCErrorCode的详细信息。

当SDK运行出现网络、媒体相关等错误时，SDK无法自动恢复，需要App干预或进行用户提示。

枚举	错误码	描述	错误原因
HRTC_ERR_CODE_SUCCESS	0	成功	-
HRTC_ERR_CODE_SDK_INTERNAL_ERROR	90000001	SDK内部系统错误	SDK内部异常。
HRTC_ERR_CODE_MSG_TOOLARGE	90000002	发送的消息太大	发送消息时，消息体太大。
HRTC_ERR_CODE_MEM_NOT_ENOUGH	90000003	内存不足	内存申请不到。
HRTC_ERR_CODE_SYNSEND_MSG_ERR	90000004	消息发送失败	消息队列异常，导致内部消息发送失败。

枚举	错误码	描述	错误原因
HRTC_ERR_CODE_PARAM_ERROR	90000005	参数错误	包括如下两方面： • 接口入参无效。 • 内部参数错误。
HRTC_ERR_CODE_API_CALLED_IN_WRONG_ORDER	90000006	API接口调用顺序不当	当前只有日志设置必须在初始化之前。
HRTC_ERR_CODE_SETUP_LOCAL_VIEW_FAIL	90000007	设置本地窗口失败	该错误码仅Android平台返回。
HRTC_ERR_CODE_START_REMOTE_STREAM_VIEW_FAIL	90000008	设置远端窗口失败	publisher场景没有远端画面，不应该设置；用户不存在；处于离会中等。
HRTC_ERR_CODE_SET_DEVICE_FAIL	90000009	设置设备失败	设置播放、录音、视频设备失败。
HRTC_ERR_CODE_INITIALIZING	90000010	初始化过程中	初始化过程中，不能再做初始化或者去初始化操作。
HRTC_ERR_CODE_UN_INITIALIZING	90000011	去初始化过程中	去初始化过程中，不能再做初始化或者去初始化操作。
HRTC_ERR_CODE_LOG_UPLOADING	90000012	日志正在上传	日志正在上传过程中。
HRTC_ERR_CODE_MEDIA_PORT_ERROR	90000013	媒体端口获取失败	音频从10010开始，视频从10020开始，尝试10次，端口都被占用了。
HRTC_ERR_CODE_WATCH_VIEW_TOO_MUCH	90000014	视频选看超过规格	当前支持最多设置16个设置远端窗口，若超过，则会失败。
HRTC_ERR_CODE_MEDIA_CMP_ERR	90000015	媒体协商失败	与服务器之间媒体协商失败。
HRTC_ERR_CODE_SERVER_NO_RESPONSE	90000016	服务器没有响应	选看在2-4s内没有收到服务器的响应。
HRTC_ERR_CODE_USER_ROLE_CHANGE_FAIL	90000017	角色切换失败	角色切换失败。
HRTC_ERR_CODE_JOIN_ROOM_FAIL	90000018	加入房间失败	加入房间失败。
HRTC_ERR_CODE_JOIN_ROOM_STATUS_BUSY	90000019	加入房间失败	已在房间中或正在网络探测中。

枚举	错误码	描述	错误原因
HRTC_ERR_CODE_JOIN_ROOM_SERVER_ERROR	90000020	加入房间失败	加入房间失败，服务器异常。
HRTC_ERR_CODE_JOIN_ROOM_SERVICE_UNREACHABLE	90000021	加入房间失败	加入房间失败，服务不可达。
HRTC_ERR_CODE_JOIN_ROOM_AUTH_FAIL	90000022	加入房间失败	加入房间失败，鉴权失败。
HRTC_ERR_CODE_JOIN_ROOM_AUTH_RETRY	90000023	加入房间失败	加入房间失败，鉴权重试。
HRTC_ERR_CODE_JOIN_ROOM_CLOCK_SYNC	90000024	加入房间失败	加入房间失败，时钟同步。
HRTC_ERR_CODE_JOIN_ROOM_URL_NOT_RIGHT	90000025	加入房间失败	加入房间失败，url错误。
HRTC_ERR_CODE_KICKED_OFF	90000026	被踢出房间	相同用户ID等原因，被踢出房间。
HRTC_ERR_CODE_SCREEN_CAPTURE_FAIL	90000027	共享失败	房间内已经存在辅流等原因，导致共享失败。
HRTC_ERR_CODE_EXT_MEDIA_OUTPUT	90000028	设置输出设备错误	当开启媒体数据输出时，不允许设置输出设备，否则会报此异常。
HRTC_ERR_CODE_RECONNECT_FAILED	90000029	连接异常	重连失败。
HRTC_ERR_CODE_SERVER_BREAK_DOWN	90000030	服务器异常	服务器宕机。
HRTC_ERR_CODE_SIGNATURE_EXPIRED	90000031	签名过期	签名已过期。
HRTC_ERR_CODE_SET_REMOTE_RENDER_MODE_FAIL	90000032	设置视图模式失败	设置远端窗口模式失败。
HRTC_ERR_CODE_SET_REMOTE_AUDIO_MUTE_FAIL	90000033	设置远端音频接收失败	订阅或取消订阅音频失败。
HRTC_ERR_CODE_SET_USEROLE_NOT_ALLOWED	90000036	跨房后，不允许在本房间内做角色切换	跨房后，不允许在本房间内做角色切换，通过**onError**返回。

枚举	错误码	描述	错误原因
HRTC_ERR_CODE_EXT_MEDIA_CAPTURE_INPUT	90000037	当前为第三方采集模式，禁用该操作	开启第三方采集模式后，禁用部分操作。
HRTC_ERR_CODE_SET_EXTAUDIO_CAPTURE_FAIL	90000038	设置第三方音频采集失败	设置第三方音频采集失败。
HRTC_ERR_CODE_SET_EXTVIDEO_CAPTURE_FAIL	90000039	设置第三方视频采集失败	设置第三方视频采集失败。
HRTC_ERR_CODE_SET_SHARE_COMPUTER_SOUND_FAIL	90000040	设置共享声音开关失败	设置共享声音开关失败。
HRTC_ERR_CODE_SET_LOCAL_AUDIO_MUTE_FAIL	90000041	启停上行音频流失败	启停上行音频流失败。
HRTC_ERR_CODE_SET_LOCAL_VIDEO_MUTE_FAIL	90000042	启停上行视频流失败	启停上行视频流失败。
HRTC_ERR_CODE_USER_REMOVED	90000043	用户被移除	用户被移除。
HRTC_ERR_CODE_ROOM_DISMISSED	90000044	房间被解散	房间被解散。
HRTC_ERR_CODE_SETUP_REMOTE_VIEW_FAIL	90000045	设置远端View失败	设置远端View失败。
HRTC_ERR_CODE_REGION_NOT_COVERED	90000048	区域未覆盖	所在区域不能提供SparkRTC服务。
HRTC_ERR_CODE_SET_EXTDATA_CAPTURE_FAIL	90000049	设置第三方辅流采集失败	设置第三方辅流采集失败
HRTC_ERR_CODE_NOT_SUPPORT	90000050	该平台不支持此功能	该平台不支持此功能
HRTC_ERR_CODE_AUDIO_ROUTE_HANDLER_NOT_INIT	90000051	音频路由没有初始化	音频路由没有初始化
HRTC_ERR_CODE_AUDIO_ROUTE_PLUGIN_CONNECTED	90000052	音频路由外设连接	音频路由外设连接
HRTC_ERR_CODE_AUDIO_ROUTE_NO_NEED_CHANGE	90000053	音频设置路由和当前路由相同	音频设置路由和当前路由相同

枚举	错误码	描述	错误原因
HRTC_ERR_CODE_AUDIO_ROUTE_CHANGE_ERROR	90000054	音频路由切换失败	音频路由切换失败
HRTC_ERR_CODE_AUDIO_ROUTE_CLOSED	90000055	音频路由控制开关关闭，无法切换路由	音频路由控制开关关闭，无法切换路由
HRTC_ERR_CODE_LOCAL_AUDIO_DISABLE_FAIL	90000056	当前未推音频流	当前未推音频流
HRTC_ERR_CODE_ROLE_NOT_SUPPORT	90000057	当前角色不支持该操作	当前角色不支持该操作
HRTC_ERR_CODE_ENABLE_BACKGROUND_FAIL	90000058	没有动态加载ML图像分割库，不能支持背景虚化和背景替换能力	没有动态加载ML图像分割库，不能支持背景虚化和背景替换能力
HRTC_ERR_CODE_ENABLE_BACKGROUND_FAIL	90000059	背景虚化或背景替换开启失败	背景虚化或背景替换开启失败
HRTC_ERR_CODE_COMMAND_REMOTEUSER_FAIL	90000060	发送CMD时，目标用户不存在	发送CMD时，目标用户不存在
HRTC_ERR_CODE_COMMAND_NOTENABLED	90000061	CMD没有被启用	CMD没有被启用
HRTC_ERR_CODE_MSG_TOOQUICK	90000062	发送的消息太频繁	发送的消息太频繁
HRTC_ERR_CODE_MSG_API_CALL_UNREASONABLE	90000063	API调用不合理	API调用不合理
HRTC_ERR_CODE_VIDEO_BAD_STATE	90000064	模块状态错误	模块状态错误
HRTC_ERR_CODE_AUDIO_SHARE_FAIL	90000065	声音共享失败	声音共享失败

7.4.10 服务端错误码

当SDK运行出现网络、媒体相关等错误时，SDK无法自动恢复，需要APP干预或进行用户提示。该错误码由服务端产生，通过**onError**返回。

表 7-26 服务端错误码

错误码	描述	错误原因
RTC.10000001	内部错误	程序或环境问题
RTC.31000000	节点不存在	程序或环境问题
RTC.31000001	session校验失败	程序或环境问题
RTC.31000003	内部异常	程序或环境问题
RTC.31000004	认证失败	用户使用问题
RTC.31000005	请重试	用户使用问题
RTC.31000006	需要时钟同步	用户使用问题
RTC.31000007	请求资源不存在	程序或环境问题
RTC.32000000	服务异常	程序或环境问题
RTC.32000001	流号已满	程序或环境问题
RTC.32000002	SFU为空	程序或环境问题
RTC.32000004	下发流信息失败	程序或环境问题
RTC.32000005	添加适配器失败	程序或环境问题
RTC.32000006	添加路由失败	程序或环境问题
RTC.32000007	获取用户信息异常	程序或环境问题
RTC.32000010	选看用户不存在	程序或环境问题
RTC.32000011	音频速率参数非法	程序或环境问题
RTC.32000012	用户列表为空	程序或环境问题
RTC.32000013	非法请求参数	程序或环境问题
RTC.32000015	内部调用异常	程序或环境问题
RTC.32000016	内部调用异常	程序或环境问题
RTC.32000017	站点不存在	程序或环境问题
RTC.32000018	错误的加密算法	程序或环境问题
RTC.32000019	客户端媒体加密密钥base64解码失败	程序或环境问题
RTC.32000020	生成媒体加密密钥失败	程序或环境问题
RTC.32000021	下发加密信息异常	程序或环境问题
RTC.32000022	获取SFU的IP失败	程序或环境问题
RTC.32000024	内部调用异常	程序或环境问题
RTC.32000025	内部调用异常	程序或环境问题

错误码	描述	错误原因
RTC.32000028	不支持的操作	程序或环境问题
RTC.32000030	sfu资源不足	程序或环境问题
RTC.32000032	跨房数量超过上限	用户使用问题
RTC.32000033	不允许重复跨入同一房间	用户使用问题
RTC.32000034	稍后重试	程序或环境问题
RTC.33000000	服务异常	程序或环境问题
RTC.33000001	节点不存在	程序或环境问题
RTC.34000001	房间已满	用户使用问题
RTC.34000002	房间不存在	程序或环境问题
RTC.34000003	站点不存在	程序或环境问题
RTC.34000004	内部调用异常	程序或环境问题
RTC.34000006	用户不存在	用户使用问题
RTC.34000007	已存在辅流共享	用户使用问题
RTC.200001	请求参数不正确	用户使用问题
RTC.200030	没有可用的mcu资源	云测问题
RTC.200031	创建推流任务失败	云测问题
RTC.200032	混流任务已存在	用户使用问题
RTC.200033	混流任务不存在	用户使用问题
RTC.200034	参与混流的用户为空	用户使用问题

7.4.11 HRTC 码率帧率配置推荐

7.4.12 数据类型

本章节列出了全平台C++ SDK的所有数据类型，您可以结合IHRTCEngine接口和回调进行开发。

表 7-27 数据类型

类型	描述
HRTCLogConfig	日志信息
HRTCUserInfo	用户信息

类型	描述
HRTCEncryptionConfig	端到端加密参数
HRTCCameraConfig	相机参数
HRTCJoinParam	入会参数
HRTCDeviceInfo	设备信息
HRTCStatsInfo	卡顿统计信息
HRTCVideoEncParam	视频编码分辨率
HRTCLocalVideoStats	本地视频流信息
HRTCRemoteVideoStats	远端视频流信息
HRTCLocalAudioStats	本地音频流信息
HRTCRemoteAudioStats	远端音频流信息
HRTCConnectInfo	跨房信息
HRTCFrameBuffer	媒体数据
HRTCVideoFrame	视频帧
HRTCAudioFrame	音频帧
HRTCVolumeInfo	发言人音量信息
HRTCNetworkTestConfig	网络探测参数配置
HRTCNetworkTestResultParam	网络探测结果参数
HRTCNetworkTestResult	网络探测结果
HRTCStreamType	流类型
HRTCVideoStreamType	大小流模式流类型
HRTCVideoDisplayMode	图像填充模式
HRTCMediaType	媒体类型
HRTCRoleType	用户角色
HRTCLogLevel	日志级别
HRTCConnStateTypes	网络连接状态
HRTCConnChangeReason	网络状态变化原因
HRTCDeviceType	系统音视频设备设备类型
HRTCDeviceState	系统音视频设备设备状态
HRTCLeaveReason	离开房间原因
HRTCVideoImageFormat	视频帧图片存储格式

类型	描述
HRTCVideoImageBufferType	视频帧缓冲区存储类型
HRTCImageBufferFormat	视频帧图片格式
HRTCAudioFrameType	音频帧格式
HRTCRemoteAudioStreamState	远端音频状态
HRTCRemoteAudioStreamStateReason	远端音频状态变化原因
HRTCRemoteVideoStreamState	远端视频状态
HRTCRemoteVideoStreamStateReason	远端视频状态变化原因
HRTCVideoMirrorType	镜像模式
HRTCMediaDirection	数据源方向
HRTCNetworkTestState	网络探测状态之成功与否
HRTCNetworkQualityLevel	网络质量信号等级
HRTCConstant	常量说明
HRTCRotationParam	摄像头参数
HRTCSpeakerModel	声音播放模式
HRTCVideoRemoteView	远端流视图
HRTCVideoRotation	视频流旋转角度
HRTCVideoOrientation	方向（横竖屏）
HRTCScreenShareIconType	捕获的共享屏幕图像类型
HRTCScreenShareSourceInfo	捕获的共享屏幕资源信息
HRTCSrceenShareOptionalInfo	其他共享屏幕的可选补充信息
HRTCScreenShareType	共享类型
HRTCRect	区域共享的自定义位置（要求有效矩形）
HRTCVideoAuxiliaryEncParam	辅流编码参数
HRTCLocalAudioStreamState	本地音频状态
HRTCLocalAudioStreamStateReason	本地音频状态变化原因
HRTCLocalVideoStreamState	本地视频状态
HRTCLocalVideoStreamStateReason	本地本地视频状态变化原因
HRTCQualityInfo	网络质量信息

类型	描述
HRTCMediaConnStateTypes	媒体连接状态类型
HRTCMediaConnChangeReason	媒体连接状态改变原因
HRTCRemoteAudioMode	远端音频模式
HRTCVideoEncodeResolutionMode	视频编码分辨率比例模式
HRTCEngineConfig	引擎初始化配置
HRTCAreaCode	访问区域
HRTCOnStats	统计回调
HRTCAudioFileState	音频播放状态
HRTCAudioFileReason	音频播放状态改变原因
HRTCMediaOptions	音视频自动选看和订阅选项
HRTCSfuType	SFU类型
HRTCRtmpUrlList	rtmp推流url列表
HRTCTranscodeConfig	rtmp推流参数结构体
HRTCUrlStatusList	rtmp推流回调url状态列表
HRTCModelType	模型类型
HRTCAudioDeviceTestVolumeNotify	音频设备测试回调数据
HRTCShareSourceInfoChangedType	正在共享的目标发生改变回调数据
HRTCSize	获取的缩略图数据分辨率
HRTCNetworkBandwidth	带宽设置参数
HRTCRemoteMicState	麦克风设备状态
HRTCMultiRoomMediaRelayConfiguration	跨房配置
HRTCSrcMultiRoomMediaInfo	源房间信息
HRTCDstMultiRoomMediaInfo	目标房间信息
HRTCMultiRoomMediaRelayState	跨房状态
HRTCMultiRoomMediaRelayStateCode	跨房状态码

HRTCLogConfig

表 7-28 日志信息

函数&属性	描述
HRTCLogLevel level	日志级别，具体请参见**HRTCLogLevel**。 • ERROR：错误级别日志 • WARNING：警告级别日志 • INFO：信息级别日志 • DEBUG：调试级别日志 默认级别为DEBUG。
const char* path	日志路径，长度不超过MAX_LOG_PATH_LEN = 226。默认路径为当前程序目录“\\rtcLog”。

HRTCUserInfo

表 7-29 用户信息

函数&属性	描述
HRTCRoleType roleType	用户角色，具体请参见**HRTCRoleType**。
const char* userId	用户ID，用户标识，长度不超过HRTC_MAX_USERID_LEN，支持64个字节的大小写字母、数字、下划线（_）中划线（-）随机组合。
const char* userName	用户昵称，用户标识，长度不超过HRTC_MAX_USERNAME_LEN。
const char* signature	鉴权签名字串。
long long ctime	UTC时间戳， 单位：秒。
const char* optionInfo	预留字段，optionalInfo是一个KV的JSON字串，可选。例如：[{key:param1,value:value1},{key:param2,value:value2}]

HRTCEncryptionConfig

表 7-30 相机参数

函数&属性	描述
cryptionMode	HRTCCryptionMode，加密模式

函数&属性	描述
suiteType	HRTCSuiteType，加密算法，仅模式HRTC_CRYPTO_AUTHENTICATION_SDK需要
cryptionSec	char *,加密密钥，仅模式HRTC_CRYPTO_AUTHENTICATION_SDK需要设置。必须是字符长度大于等于32位的16进制字符串。
secFormat	HRTCCryptionSecFormat，密钥格式，当前只支持16进制字符串。

HRTCCryptionSecFormat

表 7-31 加密模式

枚举值	描述
HRTC_HEX_STRING	16进制字符串格式。当前只支持此格式。

HRTCCryptionMode

表 7-32 加密模式

枚举值	描述
HRTC_CRYPTO_DEFAULT	不开启端到端加密，此时srtp认证（包校验）+加密。
HRTC_CRYPTO_AUTHENTICATION_SDK	开启端到端加密，srtp只认证（包校验），sdk内部加密，必须配置key。
HRTC_CRYPTO_AUTHENTICATION_APP	开启端到端加密，srtp只认证（包校验），应用层加密，需注册回调。

HRTCCameraConfig

表 7-33 相机参数

函数&属性	描述
HRTCCameraDirection direction	相机相机方向，移动端有效，前置，后置摄像头

HRTCCameraDirection

表 7-34 相机方向

枚举值	描述
HRTC_CAMERA_REAR	后置摄像头。
HRTC_CAMERA_FRONT	前置摄像头。

HRTCJoinParam

表 7-35 入会参数

函数&属性	描述
HRTCRoleType userRole	用户角色，具体请参见**HRTCRoleType**。
const char* userId	用户ID，用户标识，长度不超过HRTC_MAX_USERID_LEN，支持64个字节的大小写字母、数字、下划线（_）中划线（-）随机组合。
const char* userName	用户昵称，用户标识，长度不超过HRTC_MAX_USERNAME_LEN。
const char* authorization	签名，鉴权私钥请在**应用管理**中获取，签名的具体生成方法请参见**接入鉴权**。支持最大长度为1024。
long long ctime	UTC时间戳， 单位：秒。
const char* optionInfo	预留字段，optionalInfo是一个KV的JSON字串，可选。例如，[{key:param1,value:value1},{key:param2,value:value2}]
const char* roomId	房间号，支持最大长度64，支持数字、字母大小写、下划线、中线字符。
bool autoSubscribeAudio	是否自动订阅音频。
bool autoSubscribeVideo	是否自动订阅视频。
HRTCRemoteAudioMode scenario	使用的场景，具体请参见**HRTCRemoteAudioMode**。 • 0：主动订阅（默认）。 • 1：TopN（千人）。 • 2：P2P。 • 3：RTSA CMD自动订阅。

HRTCDeviceInfo

表 7-36 设备信息

函数&属性	描述
char deviceName[HRTC_MAX_DEVICE_NAME_LEN+ 1]	设备名称
char deviceId[HRTC_MAX_DEVICE_ID_LEN+ 1]	设备id

HRTCStatsInfo

表 7-37 卡顿统计信息

函数&属性	描述
long long mildlyFrozenCounts	600ms卡顿次数。
long long severelyFrozenCounts	超过1s卡顿次数。
long long totalMildlyFrozenTime;	600ms卡顿总时长。
long long totalSeverelyFrozenTime	1s卡顿总时长。
long long totalActiveTime	总时间，包括每一路选看时间总和。

HRTCVideoEncParam

表 7-38 视频编码分辨率

函数&属性	描述
HRTCStreamType streamType;	视频编码分辨率选择。具体请参见**HRTCStreamType**。
int width;	视频宽，根据**HRTCStreamType**和**不同分辨率下码率帧率推荐值**设置需要的分辨率和宽高比
int height;	视频高，根据**HRTCStreamType**和**不同分辨率下码率帧率推荐值**设置需要的分辨率和宽高比
int frameRate;	视频帧率，可以参考**不同分辨率下码率帧率推荐值**和**7.4.12-表14 不同场景下帧率和码率的推荐值**进行设置

函数&属性	描述
int minFrameRate;	视频最小帧率，大于0，小于frameRate
int bitrate;	视频码率，可以参考不同分辨率下码率帧率推荐值和7.4.12-表14 不同场景下帧率和码率的推荐值进行设置
int minBitrate;	视频最小码率，大于0，小于bitrate
bool disableAdjustRes;	表示上行流是否分辨率自适应，推荐开启自适应（即disableAdjustRes赋false）

表 7-39 不同分辨率下帧率和码率的推荐值

分辨率	分辨率类型	比例	最小帧率（fps）	最大帧率（fps）	最小码率	最大码率
320 X 180	SD	16:9	10	30	80	600
480 X 270	HD	16:9	10	30	160	1050
640 X 360	HD	16:9	10	30	200	1700
800 X 450	FHD	16:9	10	30	300	2100
960 X 540	FHD	16:9	10	30	400	2400
1120 X 630	FHD	16:9	10	30	450	2800
1280 X 720	FHD	16:9	10	30	500	4000
120 X 90	LD	4:3	10	30	64	240
160 X 120	SD	4:3	10	30	64	270
240 X 180	SD	4:3	10	30	80	450
320 X 240	HD	4:3	10	30	100	600
400 X 300	HD	4:3	10	30	200	900
480 X 360	HD	4:3	10	30	200	1000
640 X 480	FHD	4:3	10	30	250	1800
960 X 720	FHD	4:3	10	30	450	3000

表 7-40 不同场景下帧率和码率的推荐值

分辨率	推荐帧率	通信场景推荐码率	直播场景推荐码率
320 X 180	15	200	400

分辨率	推荐帧率	通信场景推荐码率	直播场景推荐码率
480 X 270	15	350	700
640 X 360	15	450	900
640 X 360	30	850	1700
800 X 450	15	700	1400
800 X 450	30	1050	2100
960 X 540	15	850	1700
960 X 540	30	1200	2400
1120 X 630	15	950	1900
1120 X 630	30	1400	2800
1280 X 720	15	1200	2400
1280 X 720	30	2000	4000
120 X 90	15	80	160
160 X 120	15	90	180
240 X 180	15	150	300
320 X 240	15	200	400
400 X 300	15	300	600
480 X 360	15	350	700
480 X 360	30	500	1000
640 X 480	15	600	1200
640 X 480	30	900	1800
960 X 720	15	1000	2000
960 X 720	30	1500	3000

HRTCLocalVideoStats

表 7-41 本地视频流信息详情

属性	描述
int width	视频宽
int height	视频高

属性	描述
int bitRate	视频码率
int frameRate	视频帧率，单位：fps
int packetLoss	视频丢包率
int delay	时延，单位：ms
int jitter	抖动
int sendFrameRate	实际发送帧率，单位：fps

HRTCRemoteVideoStats

表 7-42 远端视频流信息详情

属性	描述
char userId[HRTCConstant::HRTC_MAX_USERID_LEN + 1];	远端用户ID
int width	视频宽
int height	视频高
int bitRate	视频码率
int frameRate	视频帧率，单位：fps
int packetLoss	视频丢包率
int delay	时延，单位：ms
int jitter	抖动
int rendererOutputFrameRate	渲染帧率，单位：fps
int totalFrozenTime	远端用户在加入房间后到离开房间前，发生视频卡顿的累计时长，单位：ms
int frozenRate	远端用户在加入房间后到离开房间前，发生视频卡顿的累计时长占视频总有效时长的百分比

HRTCLocalAudioStats

表 7-43 本地音频流信息详情

属性	描述
int sampleRate	音频采样率

属性	描述
int channels	音频频道数
int bitRate	音频码率
int packetLoss	音频丢包率
int delay	时延，单位：ms
int jitter	抖动

HRTCRemoteAudioStats

表 7-44 远端音频流信息详情

属性	描述
char userId[HRTCConstant::HRTC_MAX_USERID_LEN + 1]	远端用户ID
int sampleRate	音频采样率
int channels	音频频道数
int bitRate	音频码率
int packetLoss	音频丢包率
int delay	时延，单位：ms
int jitter	抖动
int totalFrozenTime	远端用户在加入房间后到离开房间前，发生音频卡顿的累计时长，单位：ms
int frozenRate	远端用户在加入房间后到离开房间前，发生音频卡顿的累计时长占音频总有效时长的百分比

HRTCConnectInfo

表 7-45 跨房信息参数

函数&属性	描述
char roomId[HRTCConstant::HRTC_MAX_ROOMID_LEN + 1]	跨房房间号
HRTCRoleType role	跨房时角色，具体请参见**HRTCRoleType**。

HRTCFrameBuffer

表 7-46 媒体数据详情

函数&属性	描述
unsigned char* buffer	媒体数据数组
unsigned int bufferSize	每帧媒体数据长度

HRTCVideoFrame

表 7-47 视频帧详情

函数&属性	描述
HRTCVideoImageFormat format	视频格式，具体请参见 **HRTCVideoImageFormat**。
unsigned int width	视频宽
unsigned int height	视频高
unsigned char* data	每帧视频数据
unsigned int dataLen	每帧视频数据大小。例如，I420格式数据，dataLen = 1.5 * width * heigh。

HRTCAudioFrame

表 7-48 音频帧详情

函数&属性	描述
HRTCAudioFrameType frameType	音频格式，具体请参见 **HRTCVideoImageFormat**。
int sampleRate	音频采样率
int samplesPerSec	每秒采样数
int bytesPerSample	每个采样点占用字节数
int channels	声道数
unsigned char* data	音频数据
unsigned int dataLen	音频数据大小

HRTCVolumeInfo

表 7-49 发言人音量信息

函数&属性	描述
char userId[HRTCConstant::HRTC_MAX_USERID_LEN + 1]	远端用户ID
unsigned int volume;	音量

HRTCNetworkTestConfig

表 7-50 网络探测结果参数

函数&属性	描述
char* userId;	网络探测，用户。
char* roomId;	网络探测，房间号，建议用userId+随机数。
char* signature;	网络探测，鉴权签名字串。
long long ctime;	网络探测，UTC时间戳， 单位：秒。
int enableUplinkTest;	网络探测，开启上行探测。
int enableDownlinkTest;	网络探测，开启下行探测。
unsigned int expectedUplinkBitrate;	用户期望的最高发送码率，单位为bps，范围为0以及[100000, 5000000]，设为0表示由SDK指定最高码率。
unsigned int expectedDownlinkBitrate;	用户期望的最高接收码率，单位为bps，范围为0以及[100000, 5000000]，设为0表示由SDK指定最高码率。

HRTCNetworkTestResultParam

表 7-51 网络探测结果参数

函数&属性	描述
int bitRate;	码率
int packetLoss;	丢包
int delay;	延时
int jitter;	抖动

HRTCNetworkTestResult

表 7-52 网络探测结果

枚举值	描述
HRTCNetworkTestState	网络探测状态，具体请参见 **HRTCNetworkTestState**。
HRTCNetworkTestResultParam	上行流测试结果，具体请参见 **HRTCNetworkTestResultParam**。
HRTCNetworkTestResultParam	下行流测试结果，具体请参见 **HRTCNetworkTestResultParam**。

HRTCStreamType

表 7-53 选看分辨率

枚举值	描述
HRTC_STREAM_TYPE_SD	标清
HRTC_STREAM_TYPE_HD	高清
HRTC_STREAM_TYPE_FHD	全高清
HRTC_STREAM_TYPE_THD	真高清

HRTCVideoStreamType

表 7-54 大小流模式选看流类型

枚举值	描述
HRTC_VIDEO_STREAM_TYPE_BIG	大小流模式选看分辨率：大流类型。
HRTC_VIDEO_STREAM_TYPE_SMALL	大小流模式选看分辨率：小流类型。

HRTCVideoDisplayMode

表 7-55 图像填充模式

枚举值	描述
HRTC_VIDEO_DISPLAY_MODE_FIT	黑边模式，通过填充黑边的方式保持宽高比。
HRTC_VIDEO_DISPLAY_MODE_HIDDEN	裁剪模式，通过裁剪的方式保持宽高比。
HRTC_VIDEO_DISPLAY_MODE_FILL	缩放模式，缩放和拉伸视频尺寸以充满显示视窗。

HRTCMediaType

表 7-56 媒体类型

枚举值	描述
HRTC_MEDIA_TYPE_AUDIO	音频流。暂不支持。
HRTC_MEDIA_TYPE_VIDEO	视频流。

HRTCRoleType

表 7-57 用户角色

枚举值	描述
HRTC_ROLE_TYPE_JOINER	joiner角色，双向流角色，例如主播加入。
HRTC_ROLE_TYPE_PLAYER	player角色，接收流角色，例如观众。

HRTCLogLevel

表 7-58 日志级别

枚举值	描述
HRTC_LOG_LEVEL_ERROR	输出ERROR级别日志。
HRTC_LOG_LEVEL_WARNING	输出WARNING级别日志。
HRTC_LOG_LEVEL_INFO	输出INFO级别日志。
HRTC_LOG_LEVEL_DEBUG	输出DEBUG级别日志。

HRTCConnStateTypes

表 7-59 网络连接状态

枚举值	描述
HRTC_CONN_DISCONNECTED	连接断开。
HRTC_CONN_CONNECTING	建立网络连接中。
HRTC_CONN_CONNECTED	网络连接成功。
HRTC_CONN_RECONNECTING	重新建立网络连接中。
HRTC_CONN_FAILED	网络连接失败。

HRTCConnChangeReason

表 7-60 网络状态变化原因

枚举值	描述
HRTC_CONN_CHANGED_CONNECTING	正在连接。
HRTC_CONN_CHANGED_JOIN_SUCCESS	加入房间成功。
HRTC_CONN_CHANGED_RECONNECTING	重连中。
HRTC_CONN_CHANGED_RECONNECT_SUCCESS	重连成功。
HRTC_CONN_CHANGED_JOIN_FAILED	加入房间失败。
HRTC_CONN_CHANGED_RECONNCET_FAILED	重连失败。
HRTC_CONN_CHANGED_INTERRUPTED	连接中断。
HRTC_CONN_CHANGED_KEEP_ALIVE_TIMEOUT	心跳超时。
HRTC_CONN_CHANGED_LEAVE_ROOM	主动离开房间。
HRTC_CONN_CHANGED_JOIN_ROOM_SERVER_ERROR	服务器异常。
HRTC_CONN_CHANGED_SFU_BREAKDOWN	sfu服务故障。
HRTC_CONN_CHANGED_JOIN_ROOM_AUTH_FAILED	鉴权失败，appid或者签名错误。
HRTC_CONN_CHANGED_JOIN_ROOM_AUTH_RETRY	鉴权重试。
HRTC_CONN_CHANGED_JOIN_ROOM_AUTH_CLOCK_SYNC	时钟同步。

枚举值	描述
HRTC_CONN_CHANGED_JOIN_ROOM_URL_NOT_RIGHT	URL错误400。
HRTC_CONN_CHANGED_JOIN_ROOM_SERVICE_UNREACHABLE	服务不可达503。
HRTC_CONN_CHANGED_INTERNAL_ERROR	内部错误。
HRTC_CONN_CHANGED_KICKED_OFF	被踢出房间。
HRTC_CONN_CHANGED_SIGNATURE_EXPIRED	签名过期。
HRTC_CONN_REASON_USER_REMOVED	用户移除。
HRTC_CONN_REASON_ROOM_DISMISSED	房间解散。
HRTC_CONN_CHANGED_REGION_NOT_COVERED	区域未覆盖，所在区域不能提供SparkRTC服务。

HRTCDeviceType

表 7-61 系统音视频设备类型

枚举值	描述
HRTC_DEVTYPE_AUDIO_PLAYBACK	音频播放设备。
HRTC_DEVTYPE_AUDIO_RECORDING	音频录制设备。
HRTC_DEVTYPE_VIDEO_CAPTURE	视频采集设备。

HRTCDeviceState

表 7-62 系统音视频设备状态

枚举值	描述
HRTC_DEVICE_STATE_ACTIVE	激活状态，设备可用。
HRTC_DEVICE_STATE_DISABLED	设备禁用。
HRTC_DEVICE_STATE_UNPLUGGED	设备拔出。

HRTCLeaveReason

表 7-63 离开房间原因

枚举值	描述
HRTC_LEAVE_REASON_USER_LEAVE_ROOM	用户主动离开。
HRTC_LEAVE_REASON_SERVER_ERROR	服务器异常。
HRTC_LEAVE_REASON_BREAKDOWN	sfu服务故障。
HRTC_LEAVE_REASON_SERVICE_UNREACHABLE	服务不可达。
HRTC_LEAVE_REASON_INTERNAL_ERROR	内部错误。
HRTC_LEAVE_REASON_KICKED_OFF	被踢。
HRTC_LEAVE_REASON_SIGNATURE_EXPIRED	签名过期。
HRTC_LEAVE_REASON_RECONNECT_FAILED	重连超时。
HRTC_LEAVE_REASON_NETWORK_TEST	网络探测结束，UI不用关注。
HRTC_LEAVE_REASON_USER_REMOVED	用户移除
HRTC_LEAVE_REASON_ROOM_DISMISSED	房间解散
HRTC_LEAVE_REASON_REGION_NOT_COVERED	区域未覆盖，所在区域不能提供SparkRTC服务。

HRTCVideoImageFormat

表 7-64 视频帧图片存储格式

枚举值	描述
HRTC_VIDEO_IMAGE_FORMAT_YUV420P	YUV I420格式
HRTC_VIDEO_IMAGE_FORMAT_RGBA	RGBA格式
HRTC_VIDEO_IMAGE_FORMAT_2D	texture2d格式（仅支持Android）

HRTCVideoImageBufferType

表 7-65 视频帧缓冲区存储类型

枚举值	描述
HRTC_VIDEO_IMAGE_BUFFER_BYTE_ARRAY	Array类型，对应HRTCVideoImageFormat的YUV、RGBA格式

HRTCImageBufferFormat

表 7-66 视频帧图片格式

函数&属性	描述
HRTCVideoImageFormat format	视频帧图片存储格式
HRTCVideoImageBufferType bufferType	视频帧缓冲区存储类型

HRTCAudioFrameType

表 7-67 音频帧格式

枚举值	描述
HRTC_AUDIO_FRAME_TYPE_PCM16	PCM 16位

HRTCRemoteAudioStreamState

表 7-68 远端音频状态

枚举值	描述
HRTC_REMOTE_AUDIO_STATE_STOPPED	远端音频流关闭发送。
HRTC_REMOTE_AUDIO_STATE_STARTING	远端音频流正常编码发送。
HRTC_REMOTE_AUDIO_STATE_FIRST_DECODED	远端音频首包解码。

HRTCRemoteAudioStreamStateReason

表 7-69 远端音频状态变化原因

枚举值	描述
HRTC_REMOTE_AUDIO_REASON_REMOTE_OFFLINE	远端用户离线。
HRTC_REMOTE_AUDIO_REASON_REMOTE_MUTED	远端用户停止音频流发送。
HRTC_REMOTE_AUDIO_REASON_REMOTE_UNMUTED	远端用户开启音频流发送。
HRTC_REMOTE_AUDIO_REASON_REMOTE_FIRST_DECODED	远端音频首包解码。

HRTCRemoteVideoStreamState

表 7-70 远端视频状态

枚举值	描述
HRTC_REMOTE_VIDEO_STATE_STOPPED	远端视频流关闭发送。
HRTC_REMOTE_VIDEO_STATE_DECODING	远端视频流正常编码发送。

HRTCRemoteVideoStreamStateReason

表 7-71 远端视频状态变化原因

枚举值	描述
HRTC_REMOTE_VIDEO_REASON_REMOTE_OFFLINE	远端用户离线
HRTC_REMOTE_VIDEO_REASON_REMOTE_MUTED	远端用户停止视频流发送
HRTC_REMOTE_VIDEO_REASON_REMOTE_UNMUTED	远端用户开启视频流发送
HRTC_REMOTE_VIDEO_REASON_LOCAL_MUTED	本端已取消选看远端视频流
HRTC_REMOTE_VIDEO_REASON_LOCAL_UNMUTED	本端已开启选看远端视频流

HRTCVideoMirrorType

表 7-72 视频镜像模式

枚举值	描述
HRTC_VIDEO_MIRROR_TYPE_AUTO	移动端使用，Windows设置效果同HRTC_VIDEO_MIRROR_TYPE_DISABLE，关闭镜像。
HRTC_VIDEO_MIRROR_TYPE_ENABLE	开启镜像。
HRTC_VIDEO_MIRROR_TYPE_DISABLE	关闭镜像。

HRTCNetworkTestState

表 7-73 网络探测状态

枚举值	描述
HRTC_NETWORK_TEST_OK=0	成功
HRTC_NETWORK_TEST_FAIL	失败

HRTCMediaDirection

表 7-74 数据源方向

枚举值	描述
HRTC_MEDIA_LOCAL =0	本地数据
HRTC_MEDIA_REMOTE	远端数据

HRTCNetworkQualityLevel

表 7-75 网络质量信号等级

枚举值	描述
HRTC_NETWORK_QUALITY_UNKNOWN=0	网络质量未知
HRTC_NETWORK_QUALITY_EXCELLENT	网络质量非常好

枚举值	描述
HRTC_NETWORK_QUALITY_GOOD	网络质量好
HRTC_NETWORK_QUALITY_POOR	网络质量一般
HRTC_NETWORK_QUALITY_BAD	网络质量差
HRTC_NETWORK_QUALITY_VBAD	网络质量非常差

HRTCConstant

表 7-76 HRTCConstant 常量说明

常量	值
int HRTC_MAX_DOMAIN_LEN	119，域名长度。
int HRTC_MAX_APPID_LEN	128，app_id长度。
int HRTC_MAX_LOG_PATH_LEN	226，日志路径。
int HRTC_MAX_USERID_LEN	128，用户ID。
int HRTC_MAX_USERNAME_LEN	128，用户昵称。
int HRTC_MAX_ROOMID_LEN	128，房间号。
int HRTC_MAX_SIGNATURE_LEN	128，签名。
int HRTC_MAX_OPTION_INFO_LEN	1024，预留。
int HRTC_MAX_CODEC_NAME_LEN	64，编码详情。
int HRTC_MAX_DEVICE_NAME_LEN	256，设备名称。
int HRTC_MAX_DEVICE_ID_LEN	256，设备id。

HRTCRotationParam

表 7-77 摄像头参数

属性	描述
int captureIndex	• 0：前置摄像头 • 1：后置摄像头
int captureRotation	• 0：不旋转 • 1：逆时针旋转90度 • 2：逆时针旋转180度 • 3：逆时针旋转270度
int wndType	保留参数，当前默认设置为1
int displayRotation	• 0：不旋转 • 1：逆时针旋转90度 • 2：逆时针旋转180度 • 3：逆时针旋转270度

HRTCSpeakerModel

表 7-78 声音播放模式

枚举值	描述
HRTC_SPEAKER_MODEL_EARPIECE	耳机
HRTC_SPEAKER_MODEL_SPEAKER	扬声器

HRTCVideoRemoteView

表 7-79 远端流视图

属性	描述
void* view	窗口句柄
HRTCStreamType streamType	流模式（LD/SD/HD/FHD/THD）
const char* userId	用户ID
int disableAdjustRes	是否自适应

属性	描述
HRTCStreamType minResolution	自适应场景下，建议的最低选择档位

HRTCVideoRotation

表 7-80 视频流旋转角度

枚举值	描述
HRTC_VIDEO_ROTATION_0	不旋转
HRTC_VIDEO_ROTATION_1	逆时针旋转90度
HRTC_VIDEO_ROTATION_2	逆时针旋转180度
HRTC_VIDEO_ROTATION_3	逆时针旋转270度

HRTCVideoOrientation

表 7-81 方向（横竖屏）

枚举值	描述
HRTC_VIDEO_ORIENTATION__LANSCAPE	横屏
HRTC_VIDEO_ORIENTATION__PORTRAIT	竖屏

HRTCScreenShareIconType

表 7-82 捕获的屏幕共享图标类型

枚举值	描述
HRTC_SCREENSHARE_SMALL_ICON	小图标类型
HRTC_SCREENSHARE_BIG_ICON	大图标类型

HRTCScreenShareSourceInfo

表 7-83 共享屏幕对象信息

属性	描述
void* sourceId	采集源ID，如果是窗口共享，则为窗口句柄(HWND)
char sourceName[HRTCConstant::HRTC_MAX_SOURCE_NAME_LEN + 1]	采集源名称，如果是窗口共享，则为窗口对应名称
HRTCScreenShareType type	共享类型，具体请参见**HRTCScreenShareType**
void* icon	type是HRTC_WINDOW_CAPTURE才有效，返回类型是Windows资源类型HICON

HRTCScreenShareType

表 7-84 屏幕共享类型

属性	描述
HRTC_SCREEN_CAPTURE	屏幕共享
HRTC_WINDOW_CAPTURE	窗口共享

HRTCSrceenCaptureOptionalInfo

表 7-85 其他共享屏幕可选补充信息

属性	描述
bool disableCaptureMouse	是否禁止鼠标采集，默认false采集鼠标
HRTCRect rect	程序共享下支持区域共享，宽不能超过1920高度不能超过1080，最小宽不能小于320，高度不能小于240，共享前设置，区域要求是固定的，具体请参见**HRTCRect**

HRTCRect

表 7-86 区域共享的自定义位置

枚举值	描述
unsigned int left	自共享对象（屏幕或窗口）左上角起取的坐标，上下左右要求构成一个有效矩形（Rect），方可实现自定义区域共享
unsigned int top	
unsigned int right	
unsigned int bottom	

HRTCVideoAuxiliaryEncParam

表 7-87 辅流编码参数

属性&函数	描述
int frameRate	帧率，推荐帧率15。
int width	宽
int height	高
int bitrate	码率，可以参考**7.4.12-表13 不同分辨率下帧率和码率的推荐值**和**7.4.12-表14 不同场景下帧率和码率的推荐值**进行设置

HRTCLocalAudioStreamState

表 7-88 本地音频状态

枚举值	描述
HRTC_LOCAL_AUDIO_STATE_STOPPED	本地音频流默认初始状态
HRTC_LOCAL_AUDIO_STATE_RECORDING	本地音频流录制设备启动成功
HRTC_LOCAL_AUDIO_STATE_FAILED	本地音频流启动失败

HRTCLocalAudioStreamStateReason

表 7-89 本地音频状态变化原因

枚举值	描述
HRTC_LOCAL_AUDIO_REASON_ERROR_OK	本地音频流状态正常
HRTC_LOCAL_AUDIO_REASON_ERROR_FAILURE	本地音频流出错原因不明确
HRTC_LOCAL_AUDIO_REASON_ERROR_RECORD_FAILURE	本地音频流录制失败，建议您检查录制设备是否正常工作
HRTC_LOCAL_AUDIO_REASON_ERROR_STOP_FAILURE	关闭采集失败
HRTC_LOCAL_AUDIO_REASON_ERROR_ACCESS_DENIED	音频设备无法访问，可能是设备隐私权限设置问题
HRTC_LOCAL_AUDIO_REASON_ERROR_ON_EXCLUSIVE_MODE	音频设备处于独占模式，且被其他应用独占，可以通知用户取消独占模式
HRTC_LOCAL_AUDIO_REASON_ERROR_ENDPOINT_CREATE_FAILED	音频设备终端创建失败，音频设备被拔出，或者已重新配置，禁用，删除了音频硬件或关联的硬件资源不可用。使用其他音频设备，重启或者更新驱动（仅适用于Windows）
HRTC_LOCAL_AUDIO_REASON_ERROR_MMSYSERR_INVALPARAM	音频设备API非法参数，目前已知是杀毒软件导致（仅适用于Windows）
HRTC_LOCAL_AUDIO_REASON_ERROR_MMSYSERR_NODRIVER	音频设备API返回无驱动，需要用户升级驱动（仅适用于Windows）
HRTC_LOCAL_AUDIO_REASON_ERROR_AUDIO_SERVER_NOT_RUNNING	用户windows audio服务未启动，或者启动失败（仅适用于Windows）
HRTC_LOCAL_AUDIO_REASON_ERROR_NO_DEVICE	没有设备（仅适用于Windows）
HRTC_LOCAL_AUDIO_REASON_ERROR_RESTART_FAILED	扬声器播放无数据，重启失败

HRTCLocalVideoStreamState

表 7-90 本地视频状态

枚举值	描述
HRTC_LOCAL_VIDEO_STATE_STOPPED	本地视频流默认初始状态

枚举值	描述
HRTC_LOCAL_VIDEO_STATE_CAPTURING	本地视频流采集设备启动成功
HRTC_LOCAL_VIDEO_STATE_FAILED	本地视频流启动失败

HRTCLocalVideoStreamStateReason

表 7-91 本地视频状态变化原因

枚举值	描述
HRTC_LOCAL_VIDEO_REASON_ERROR_OK	本地视频流状态正常
HRTC_LOCAL_VIDEO_REASON_ERROR_FAILURE	本地视频流出错原因不明确
HRTC_LOCAL_VIDEO_REASON_ERROR_CAPTURE_FAILURE	本地视频流录制失败，建议您检查录制设备是否正常工作
HRTC_LOCAL_VIDEO_REASON_ERROR_STOP_FAILURE	关闭采集失败
HRTC_LOCAL_VIDEO_REASON_ERROR_CAPTURE_DEVICE_NO_PERMISSION	没有摄像头权限
HRTC_LOCAL_VIDEO_REASON_ERROR_CAPTURE_DEVICE_BUSY	摄像头设备已占用
HRTC_LOCAL_VIDEO_REASON_ERROR_CAPTURE_APP_IN_BACKGROUND	应用处于后台，仅适用Android和iOS
HRTC_LOCAL_VIDEO_REASON_ERROR_CAPTURE_OPEN_CAMERA_FAILED	打开摄像头设备失败
HRTC_LOCAL_VIDEO_REASON_ERROR_CAPTURE_MULTIPLE_FOREGROUND_APP	应用窗口处于侧拉、分屏、画中画模式（仅适用于iOS）
HRTC_LOCAL_VIDEO_REASON_ERROR_CAPTURE_DEVICE_DISCONNECTED	本地视频采集设备未连接（仅适用Windows和macOS）

HRTCQualityInfo

表 7-92 网络质量信息

属性	描述
char userId[HRTCConstant::HRTC_MAX_USERID_LEN + 1]	用户ID
int width	宽
int height	高
HRTCNetworkQualityLevel level	网络质量等级
HRTCMediaType mediaType	媒体流类型

HRTCMediaConnStateTypes

表 7-93 媒体连接状态类型

枚举值	描述
HRTC_MEDIA_CONN_CONNECTED	与媒体服务器连接成功
HRTC_MEDIA_CONN_FAILED	与媒体服务器建链失败

HRTCMediaConnChangeReason

表 7-94 媒体连接状态变化原因

枚举值	描述
HRTC_MEDIA_CONN_CHANGED_CONNECTED	连接成功
HRTC_MEDIA_CONN_CHANGED_NAT_FAILED	与媒体服务器NAT未打通

HRTCRemoteAudioMode

表 7-95 远端音频流收流模式

枚举值	描述
HRTC_REMOTE_AUDIO_SUBSCRIBED = 0	订阅模式（自主订阅）

枚举值	描述
HRTC_REMOTE_AUDIO_TOP_THREE = 1	TopN模式（收音量最大的三路流）
HRTC_REMOTE_AUDIO_P2P = 2	P2P模式
HRTC_REMOTE_AUDIO_RTSA_CMD = 3	RTSA-CMD模式

HRTCVideoEncodeResolutionMode

表 7-96 视频编码分辨率比例模式

枚举值	描述
HRTC_VIDEO_ENCODE_RESOLUTION_MODE_NONE = 0	不固定比例
HRTC_VIDEO_ENCODE_RESOLUTION_MODE_CONST_RATIO = 1	固定比例

HRTCEngineConfig

表 7-97 引擎初始化配置

属性	类型	描述
appId	const char *	应用ID，只有App ID相同的应用程序才能进入同一个房间进行互动。appId获取方法请参见**应用管理**。
countryCode	const char *	国家码，具体值请参见**国家码对照表**
enableHaTrace	bool	打点开关
muteAudioRoute	bool	是否禁音频路由
enableAudio	bool	音频模块开关
enableVideo	bool	视频模块开关
enableShare	bool	共享模块开关
audioScene	**HRTCAudioSceneType**	音频场景，会议/音乐
audioQualityLevel	**HRTCAudioQualityLevel**	音频档位，16k/48k

HRTCNetProxyConfig

表 7-98 代理参数的配置

属性	描述
bool autoNetProxy	是否开启自动代理
const char *address	代理的地址
int port	代理端口
const char *name	代理认证的账号名
const char *pwd	代理认证的账号密码

HRTCAreaCode

表 7-99 访问区域

枚举值	描述
HRTC_AREA_CODE_GLOB	全球（默认）
HRTC_AREA_CODE_CN	中国
HRTC_AREA_CODE_NA	中北美
HRTC_AREA_CODE_SA	拉美
HRTC_AREA_CODE_EU	欧洲
HRTC_AREA_CODE_SEA	东南亚
HRTC_AREA_CODE_AF	非洲
HRTC_AREA_CODE_AS	亚洲

HRTCSetupRemoteViewResult

表 7-100 批量选看结果

属性&函数	描述
char userId[HRTCConstant::HRTC_MAX_USERID_LEN + 1]	用户ID
unsigned char result	选看结果

属性&函数	描述
int code	状态码 ● 0：成功 ● 1：失败
char reason[HRTCConstant::HRTC_MAX_WATCH_RESULT_REASON_LEN + 1]	原因描述

HRTCOnStats

表 7-101 统计信息

属性	描述
double cpuAppUsage	app的cpu利用率
double cpuTotalUsage	cpu总利用率
unsigned int memoryAppUsageInKbytes	app占用内存
double memoryAppUsageRatio	app内存占用率
double memoryTotalUsageRatio	总的内存利用率
int gatewayRtt	到本地网关的延迟
unsigned long long sendBytes	总的发送字节数
unsigned long long sendVideoBytes	视频的发送字节数
unsigned long long sendAudioBytes	音频的发送字节数
unsigned long long receiveBytes	总的接收字节数
unsigned long long receiveVideoBytes	视频的接收字节数
unsigned long long receiveAudioBytes	音频的接收字节数
unsigned int sendBitRate	总的发送比特率
unsigned int sendVideoBitRate	视频的发送比特率
unsigned int sendAudioBitRate	音频的发送比特率
unsigned int receiveBitRate	总的接收比特率
unsigned int receiveVideoBitRate	视频的接收比特率
unsigned int receiveAudioBitRate	音频的接收比特率
unsigned int sendLossRate	发送丢包率
unsigned int receiveLossRate	接收丢包率

属性	描述
unsigned int lastmileDelay	到服务器的延迟

HRTCAudioFileState

表 7-102 访问区域

枚举值	描述
HRTC_AUDIO_FILE_OPEN_COMPLETED	成功打开音频文件
HRTC_AUDIO_FILE_OPENING	正在打开音频文件
HRTC_AUDIO_FILE_IDLE	音频文件播放就绪
HRTC_AUDIO_FILE_PLAYING	音频文件播放中
HRTC_AUDIO_FILE_PLAY_COMPLETED	音频文件播放完成
HRTC_AUDIO_FILE_PAUSED	音频文件暂停播放
HRTC_AUDIO_FILE_STOPPED	音频文件停止播放
HRTC_AUDIO_FILE_FAILED	音频文件播放失败
HRTC_AUDIO_FILE_POSITION_UPDATE	音频文件播放进度更新
HRTC_AUDIO_FILE_STATE_UNKNOWN	音频文件播放状态未知

HRTCAudioFileReason

表 7-103 音频播放状态改变原因

枚举值	描述
HRTC_AUDIO_FILE_REASON_NONE	没有错误
HRTC_AUDIO_FILE_REASON_URL_NOT_FOUND	未找到URL
HRTC_AUDIO_FILE_REASON_CODEC_NOT_SUPPORTED	解码器不支持该编码
HRTC_AUDIO_FILE_REASON_INVALID_ARGUMENTS	非法参数
HRTC_AUDIO_FILE_REASON_SRC_BUFFER_UNDERFLOW	播放缓冲区数据不足
HRTC_AUDIO_FILE_REASON_INTERNAL	内部错误

枚举值	描述
HRTC_AUDIO_FILE_REASON_INVALID_STATE	播放器内部状态错误
HRTC_AUDIO_FILE_REASON_NO_RESOURCE	没有该资源
HRTC_AUDIO_FILE_REASON_OBJ_NOT_INITIALIZED	对象未初始化
HRTC_AUDIO_FILE_REASON_INVALID_CONNECTION_STATE	播放器与服务器连接无效
HRTC_AUDIO_FILE_REASON_UNKNOWN_STREAM_TYPE	未知的媒体流类型
HRTC_AUDIO_FILE_REASON_VIDEO_RENDER_FAILED	渲染失败
HRTC_AUDIO_FILE_REASON_INVALID_MEDIA_SOURCE	无效的媒体资源
HRTC_AUDIO_FILE_REASON_UNKNOWN	状态未知

HRTCMediaOptions

表 7-104 音视频自动选看和订阅选项

属性	描述
bool autoSubscribeAudio	自动订阅远端用户音频流
bool autoSubscribeVideo	自动订阅远端用户视频流

HRTCMediaOptions() {
autoSubscribeAudio = true;
autoSubscribeVideo = true;
}
	初始化构造默认true自动订阅和选看

HRTCAudioQualityLevel

表 7-105 音频档位

枚举值	描述
HRTC_AUDIO_QUALITY_LEVEL_DEFAULT	默认值，表示使用采样率16KHZ、单声道、编码码率最大值为30Kbps

HRTCAudioSceneType

表 7-106 音频场景

枚举值	描述
HRTC_AUDIO_SCENE_DEFAULT	默认值，表示会议模式
HRTC_AUDIO_SCENE_MUSIC	表示音乐模式

HRTCSfuType

表 7-107 Sfu 类型

枚举值	描述
HRTC_SFU_TYPE_PUBLIC_NETWORK	公网sfu资源

HRTCRtmpUrlList

表 7-108 HRTCRtmpUrlList

属性	描述
int nSize	数组大小
HRTCRtmpUrl urlInfo[5]	url数组，具体请参见表7-109

表 7-109 HRTCRtmpUrl

属性	描述
char url[1025]	url字符串

HRTCTranscodeConfig

表 7-110 HRTCTranscodeConfig

属性	描述
HRTCRtmpConfig config	Rtmp推流通用配置，如码率等，具体请参见表7-111
HRTCRtmpUserInfo rtmpUserInfo[50]	Rtmp推流的用户流信息，具体请参见表7-112

表 7-111 HRTCRtmpConfig

属性	描述
int width	旁路推流的输出视频流的总宽度，单位为px。默认值为360，取值范围为[64-1920]
int height	旁路推流的输出视频流的总高度，单位为px。默认值为640，取值范围为[64-1920]
int videoBitrate	旁路推流的输出视频的码率，单位为Kbps。默认值为400Kbps，取值范围为[32-2760]
int videoFramerate	旁路推流的输出视频的帧率，单位为fps。默认值为15，取值范围为[10,30]
int videoGop	用于旁路直播的输出视频的GOP，单位为帧。默认值为30帧，取值范围为[1-300]
int audioSampleRate	用于旁路直播的输出音频的采样率，默认为16000，取值范围为[16000-96000]
int audioBitrate	旁路直播的输出音频的码率，单位为Kbps。默认值为48，最大值为128，取值范围为[1-128]
int audioChannels	旁路直播的输出音频的声道数，默认为1，取值范围为[1-5]
int tmplate	0表示悬浮，1表示九宫格，2表示屏幕分享，默认为0

表 7-112 HRTCRtmpUserInfo

属性	描述
char userId[64]	用户id
bool main	是否推大流
bool slides	是否推小流
bool desktop	是否推桌面流
bool audio	是否推音频流

表 7-113 HRTCRtmpUserInfoList

属性	描述
int nSize	数组大小
HRTCRtmpUserInfo rtmpUserInfo[50]	结构体数组

HRTCUrlStatusList

表 7-114 HRTCRtmpUrlInfo

属性	描述
char url[1025]	url字符串
int status	状态码
int errCode	错误码

表 7-115 rtmp 推流回调 url 状态数组

属性	描述
int nSize	数组大小
HRTCRtmpUrlInfo rtmpUrlInfo[5]	url状态数组，具体请参见表7-112

HRTCAudioOperateMode

表 7-116 采集数据回调的处理模式

属性	描述
enum HRTCAudioOperateMode.HRTC_AUDIO_OPERATE_READ_AND_WRITE	可读可写模式

HRTCModelType

表 7-117 HRTCModelType

属性	描述
HRTC_MODEL_VOICE_ACTIVITY_DETECTION	人声检测
HRTC_MODEL_VOICE_HOWLING_SUPPRESSION	啸叫抑制

HRTCEngineContext

表 7-118 引擎初始化参数

属性	类型	描述
engineConfig	HRTCEngineConfig	引擎配置项，具体请参见**HRTCEngineConfig**。
logConfig	HRTCLogConfig	日志配置项，具体请参见**HRTCLogConfig**。
eventHandler	IHRTCEngineEventHandler	事件回调，具体请参见**IHRTCEngineEventHandler**。

HRTCLogConfig

表 7-119 日志参数

属性	类型	描述
level	HRTCLogLevel	日志级别，具体请参见**HRTCLogLevel**，默认值HRTC_LOG_LEVEL_DEBUG，推荐使用HRTC_LOG_LEVEL_DEBUG。
path	const char *	日志路径，需调用方保证路径合法可用，rtc仅做基础校验
logSize	int	日志大小，默认值10M，推荐10M
enable	bool	日志开关

HWRtcGSensorMode

表 7-120 重力感应模式

属性	描述
HWRtcGSensorModeDisable	关闭重力感应
HWRtcGSensorModeUIAutoLayout	开启重力感应 **注意**：SDK不会根据陀螺仪自动调整本地View的画面方向，而是需要您的APP开启了重力感应进行适配

HRTCBeautyOptions

表 7-121 美颜选参

属性	类型	描述
buffingLevel	float	磨皮档位，取0-1浮点数，默认值为0.5，推荐使用默认值0.5
complexionAlpha	float	肤色档位，取0-1浮点数，默认值为0.5，推荐使用默认值0.5

HRTCAudioDeviceTestVolumeNotify

表 7-122 HRTCAudioDeviceTestVolumeNotify

属性	描述
recordVolume	麦克风音量
playbackVolume	扬声器音量

HRTCShareSourceInfoChangedType

表 7-123 HRTCShareSourceInfoChangedType

属性	描述
HRTC_SHARE_SOURCE_INFO_CHANGED_DIALED_OUT	正在共享的屏幕被拔出
HRTC_SHARE_SOURCE_INFO_CHANGED_RESOLUTION_CHANGE	分辨率有变化
HRTC_SHARE_SOURCE_INFO_CHANGED_REGION_CHANGE	区域有变化
HRTC_SHARE_SOURCE_INFO_CHANGED_NUM_REDUCE	屏幕个数有减少
HRTC_SHARE_SOURCE_INFO_CHANGED_NUM_INCREASE	屏幕个数有增加
HRTC_SHARE_SOURCE_INFO_CHANGED_IS_BLOCKED	正在共享的窗口被遮挡了
HRTC_SHARE_SOURCE_INFO_CHANGED_CLOSE	正在共享的窗口关闭了
HRTC_SHARE_SOURCE_INFO_CHANGED_MINIMIZE	正在共享的窗口最小化了

属性	描述
HRTC_SHARE_SOURCE_INFO_CHANGED_VALID_AREA_LESS_THAN_96	窗口有效区域小于96

HRTCSize

表 7-124 缩略图数据分辨率

属性	类型	描述
width	unsigned int	宽
height	unsigned int	高

HRTCNetworkBandwidth

表 7-125 带宽设置参数

属性	类型	描述
maxBandwidth	int	带宽上限。单位：KB。有效范围为[3072, 51200]，即3M~50M。

HRTCMultiRoomMediaRelayConfiguration

表 7-126 跨房配置

属性	类型	描述
srcRoomMediaInfo	**HRTCSrcMultiRoomMediaInfo**	源房间的鉴权信息
destRoomMediaInfo	**HRTCDstMultiRoomMediaInfo**	目的跨房的房间信息以及鉴权信息

HRTCSrcMultiRoomMediaInfo

表 7-127 源房间信息

属性	类型	描述
authorization	const char*	源房间的鉴权信息

属性	类型	描述
userId	const char*	源房间的用户名 (必须为0)
roomId	const char*	源房间的房间号
ctime	long long	鉴权时间信息

HRTCDstMultiRoomMediaInfo

表 7-128 目标房间信息

属性	类型	描述
authorization	const char*	目标跨房的鉴权信息
userId	const char*	目标跨房的虚拟用户名
roomId	const char*	目标跨房房间号
userRole	HRTCRoleType	跨房角色
ctime	long long	鉴权时间信息

HRTCMultiRoomMediaRelayState

表 7-129 跨房状态

属性	描述
HRTC_MULTI_ROOM_MEDIA_RELAY_STATE_IDLE	就绪状态
HRTC_MULTI_ROOM_MEDIA_RELAY_STATE_CONNECTING	正在连接
HRTC_MULTI_ROOM_MEDIA_RELAY_STATE_RUNNING	主播成功加入目标房间
HRTC_MULTI_ROOM_MEDIA_RELAY_STATE_FAILURE	发生异常

HRTCMultiRoomMediaRelayStateCode

表 7-130 跨房状态码

属性	描述
HRTC_MULTI_ROOM_MEDIA_RELAY_OK	正常状态
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_SERVER_NO_RESPONSE	服务端无响应
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_INTERNAL_ERROR	服务器内部出错
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_USER_OVER_LIMIT	用户跨房超出限制数
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_OVER_LIMIT	房间跨房用户超出限制数
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_REQ_EMPTY	跨房请求消息体为空
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_OPERATION_CONFLICT	跨房请求，加入和退出存在冲突
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_SRC_USERINFO_INVALID	跨房请求原用户信息无效
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_WITH_ORI	跨房房间与原用户房间相同
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_REPEAT	跨房请求房间重复
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_USER_EXISTED	跨房用户已存在
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_INVALID_REQUEST	无效请求
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_ROOM_IS_NOT_EXIST	房间不存在
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_FRAME_TYPE_NOT_EQUAL	跨房源房间和目的房间加密模式不一致
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_AUTHENTICATION_FAILURE	鉴权失败
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_REMOVE_INFO_NOT_EXIST	退出跨房信息不存在
HRTC_MULTI_ROOM_MEDIA_RELAY_ERROR_EXCEPTION_STOP	异常退出跨房

HRTCRemoteMicState

表 7-131 麦克风设备状态

属性	描述
HRTC_REMOTE_MIC_STATE_UNMUTE	麦克风设备状态正常
HRTC_REMOTE_MIC_STATE_MUTE	麦克风设备状态静音

7.5 常见问题

- **调用加入房间接口成功后，再调用pushExternalVideoFrame返回90000001 SDK内部系统错误**

  需要收到加入房间成功回调通知后，才能调用pushExternalVideoFrame。

- **joiner和player都加入房间成功，joiner调用pushExternalVideoFrame返回0，但是player没有收到onRenderVideoFrame回调**

  需要接收端在收到onUserJoined消息后，在主线程里调用pullRemoteVideo和startRemoteStreamView。

- **startRemoteStreamView会返回90000008设置远端窗口失败错误**

  startRemoteStreamView的调用是需要放在主线程，不能在onUserJoined调用，否则会返回90000008错误码。

- **1080p经过RTC传输后，被自动缩放且颜色改变**

  老版本SDK不支持1080p。

- **推流1080p，接收到的是360p的流**

  关闭发送端设置发流编码参数分辨率自适应，关闭收端分辨率自适应。

  ```
  int setVideoEncoderConfig(unsigned int totalBitRate, HRTCVideoEncParam *encoderParams, unsigned int encoderCount) // HRTCVideoEncParam.disableAdjustRes=true
  int setVideoEncoderConfig(const HRTCVideoEncParam &encParam) // HRTCVideoEncParam.disableAdjustRes=true
  int startRemoteStreamView(const char* userId, view_t view, HRTCStreamType streamType, bool disableAdjustRes) // disableAdjustRes=true
  ```

- **rtc设置了音频的自渲染和自采集，但是不推音频的pcm数据，onPlaybackAudioFrame也会收到回调**

  这是正常现象，SDK会定时回调onPlaybackAudioFrame，里面是空白语音包。

- **自渲染没有图像**

  onVideoFrameRender回调函数实现中，y、u、v分量需要按如下方式获取：

  ```
  virtual bool onVideoFrameRender(const char* userId, HRTCVideoFrame& videoFrame)
  {
      Frame frame;
      frame.width = videoFrame.width;
      frame.height = videoFrame.height;
      frame.yBuf = videoFrame.data;
      frame.uBuf = videoFrame.data + videoFrame.width * videoFrame.height;          //宽*高
      frame.vBuf = videoFrame.data + videoFrame.width * videoFrame.height * 5 / 4; // 宽*高*5/4
      frame.yStride = videoFrame.Stride;
      frame.uStride = videoFrame.Stride / 2;
      frame.vStrides = videoFrame.Stride / 2;
      //用frame取渲染窗口
      return true;
  }
  ```

7.6 修订记录

表 7-132 修订记录

修改时间	修改说明
2022-06-21	第六次正式发布 本次变更如下： ● 新增以下接口： – addMultiRoomMediaRelay：添加单个跨房 – removeMultiRoomMediaRelay：删除单个跨房 – stopMultiRoomMediaRelay：停止所有跨房 – onMultiRoomMediaRelayStateChanged：跨房状态回调 – appendLocalView：设置本地视频另一个窗口显示 – appendRemoteView：设置远端视频另一个窗口显示 ● 新增以下事件回调： – onRenderExternalVideoFrame：视频自渲染回调 – onPlaybackExternalAudioFrame：音频自渲染回调 – onMultiRoomMediaRelayStateChanged：跨房状态改变回调 – onRemoteMicrophoneStateChanged：麦克风设备状态变更通知 – onUserNetworkQualityNotify：加入房间后的网络质量状态回调 ● 新增以下数据类型： – HWRtcRemoteMicState：麦克风设备状态 – HRTCMultiRoomMediaRelayConfiguration：跨房配置 – HRTCSrcMultiRoomMediaInfo：源房间信息 – HRTCDstMultiRoomMediaInfo：目标房间信息 – HRTCMultiRoomMediaRelayState：跨房状态 – HRTCMultiRoomMediaRelayStateCode：跨房状态码
2022-03-24	第五次正式发布 本次变更如下： 修改appid获取方式的相关描述。
2022-03-18	第四次正式发布 本次变更如下： 新增setNetworkBandwidth接口

修改时间	修改说明
2022-02-25	第三次正式发布 本次变更如下： 新增以下接口： ● recordingDeviceTest音频采集设备测试 ● finishRecordingDeviceTest结束音频采集设备测试 ● playbackDeviceTest音频播放设备测试 ● finishPlaybackDeviceTest结束音频播放设备测试 ● echoTest音频设备回路测试 ● finishEchoTest结束音频设备回路测试 ● cameraDeviceTest视频采集设备测试 ● finishCameraDeviceTest结束视频采集设备测试 ● onAudioDeviceTestVolumeNotify音频设备测试回调 ● setVideoWaterMark插入/删除水印 ● setAccessResourceType设置接入环境 ● setBackgroungBlur设置本地视频背景虚化 ● setBackgroundReplace设置本地视频背景替换 修改以下接口： HRTCVideoDisplayMode去掉自适应HRTC_VIDEO_DISPLAY_MODE_ADAPT
2021-12-02	第二次正式发布 本次变更如下： 优化部分文档描述。
2021-11-22	第一次正式发布

8 Web SDK

8.1 浏览器适配

本章节介绍Web SDK支持的浏览器类型、版本以及使用限制。

表 8-1 浏览器适配详情

操作系统类型	浏览器类型	浏览器版本	SDK版本约束	下行（播放）	上行（上麦）	屏幕分享
Windows	Chrome浏览器	67+	• v1.10.0及以上版本 • v2.0.0及以上版本	支持	支持	支持（Chrome 73+版本）
	QQ浏览器（极速内核）	10.4+				不支持
	360安全浏览器（极速模式）	12				支持
	微信内嵌浏览器	-	v2.0.0及以上版本	支持	不支持	不支持
	企业微信内嵌浏览器	-				
	Firefox浏览器	90+	v2.0.1及以上版本	支持	支持	支持
	Edge浏览器	80+	v2.0.2及以上版本	支持	支持	支持
	搜狗浏览器（高速模式）	11+				

操作系统类型	浏览器类型	浏览器版本	SDK版本约束	下行（播放）	上行（上麦）	屏幕分享
	Opera浏览器	54+				支持（ Opera 60+ 版本）
macOS	Chrome浏览器	67+	• v1.10.0及以上版本 • v2.0.0及以上版本	支持	支持	支持（Chrome 73+ 版本）
	微信内嵌浏览器	-	v2.0.0及以上版本	支持	不支持	不支持
	企业微信内嵌浏览器	-				
	Safari浏览器	11+	v2.0.1及以上版本	支持	支持	支持（ Safari 13+ 版本）
	Firefox浏览器	90+				支持
	Edge浏览器	80+				
	Opera浏览器	56+	v2.0.2及以上版本	支持	支持	支持（ Opera 60+ 版本）
Android	微信内嵌浏览器（ TBS 内核）	-	• v1.10.0及以上版本 • v2.0.0及以上版本	支持	支持	不支持
	微信内嵌浏览器（ XWEB内核）	-				
	企业微信内嵌浏览器	-				
	移动版Chrome浏览器	-				
	移动版QQ浏览器	12+				

操作系统类型	浏览器类型	浏览器版本	SDK版本约束	下行（播放）	上行（上麦）	屏幕分享
	移动版华为浏览器	11.0.8+				
	移动版UC浏览器	-	-	不支持	不支持	不支持
iOS	微信内嵌浏览器	iOS 14.3+ 微信6.5+版本	• v1.10.0及以上版本 • v2.0.0及以上版本	支持	支持	不支持
	移动版Chrome浏览器	iOS 14.3+	v2.0.0及以上版本	支持	支持	不支持
	企业微信内嵌浏览器	-			不支持	
	移动版Safari浏览器	11+	v2.0.1及以上版本	支持	支持	不支持

表 8-2 浏览器使用限制

浏览器类型	使用限制
Chrome浏览器	• 在移动端浏览器上，**getSpeakers**接口只能获取到default音频输出设备。 • 在移动端浏览器上，不支持采集120p及以下的分辨率。 • 在华为移动端设备上，Chrome浏览器（包括华为自带浏览器）支持WebRTC的版本为91+。 • 在Mac Chrome浏览器上使用屏幕分享前，需确保已在“设置 > 安全性与隐私 > 隐私 > 屏幕录制”中打开Chrome屏幕录制授权。

浏览器类型	使用限制
Safari浏览器	• 在Safari 11及12上，需要在建链之前调用navigator.mediaDevices.getUserMedia接口（调用sdk接口**createStream**创建本地流）获取媒体权限，否则媒体无法交互（媒体链路无法建立）。 • 华为Native SDK推流，在Safari 11及12浏览器中选看，存在绿屏现象。 • Safari 13的用户可能听不到远端用户的声音。 • iOS Safari 14.2和macOS Safari 14.0.1上音频可能断断续续。 • Safari 15.1发布流时会发生异常，导致页面崩溃。 • Safari不支持获取输出设备信息，因此，不支持**getSpeakers**和**setAudioOutput**接口。 • Safari不支持调用**addTrack**和**removeTrack**接口。 • Safari浏览器不能多次调用本地流采集接口，否则会引起采集黑屏，需在调用音视频采集接口前关闭前一次采集。 • Safari不支持大小流。 • iOS移动端浏览器不支持混音功能。 • Mac设备Safari 12插耳机的场景下，开启混音后，远端听不到任何声音。 • 须在选看远端用户音频前调用navigator.mediaDevices.getUserMedia接口，否则无法听到声音和无法获取音频音量值。 • macOS Ventura系统，Safari 16.1使用屏幕分享时，会出现本地屏幕共享流黑屏。
Firefox浏览器	• Firefox只支持30fps视频帧率。 • Apple M1芯片的Mac设备上Firefox不支持H.264编解码。 • Firefox不支持获取输出设备信息，因此，不支持**getSpeakers**和**setAudioOutput**接口。 • Firefox 97+浏览器推流场景，其他端选看存在黑屏、卡顿兼容问题，正在紧急修复中。请暂时使用其他浏览器。
Opera浏览器	在华为移动端设备上，Opera浏览器支持WebRTC的版本为64+。
其他浏览器	由于各设备厂家的浏览器内核、webview、版本等因素，移动端浏览器对WebRTC的支持度不一，除可以使用**表8-1**中列举的移动端浏览器类型外，还可以集成使用Native SDK（Andriod / iOS）。

⚠ 注意

- 已集成Web SDK 1.0+版本（2.0+版本不涉及）的用户，请尽快升级至1.10.0+版本，否则在Chrome 96+浏览器上有可能出现无法使用的情况。
- Web SDK 2.0+版本是目前的主力构建版本，承载新功能及体验优化，建议您优先集成使用。Web SDK 1.0+版本仅做存量用户的维护，不再构建新的功能。
- Web SDK 1.0+和Web SDK 2.0+业务上不能互通，集成时需要注意。
- Safari浏览器上的使用限制和已知问题较多，建议使用兼容性较好的Chrome浏览器或者集成Native SDK。

8.2 开发前准备

前提条件

已**提交工单**获取SDK包。

环境要求

- 编译工具推荐安装Microsoft Visual Studio Code 1.43.2或以上版本。
- 如果客户端用Node.js开发， 推荐安装14.19.1或以上版本。
- 支持的浏览器详情请参见**Web浏览器适配详情**。
- 如果客户端用TypeScript开发，TypeScript的版本不低于3.8.3。
- 由于浏览器安全策略限制，仅支持通过**https://域名**的方式访问，或者直接在本地搭建服务器，通过**localhost:端口**访问，否则无法获取摄像头及麦克风的权限。

SDK 集成

步骤1 将获取的SDK压缩包放置在自己项目的“sdk”目录下。

步骤2 在项目代码中引入“hrtc”。

- 如果您通过<script>方式引入华为WebRTC SDK，则通过访问HRTC获取导出的模块：

```
<script src='./sdk/hrtc.js'>
    console.log(HRTC.VERSION)
</script>
```

- 如果您直接引用华为WebRTC SDK静态JS文件，则通过以下方式访问：

```
import HRTC from '../../sdk/hrtc'
console.log(HRTC.VERSION)
```

- 如果您通过npm模块化的方式引入华为WebRTC SDK，首先要安装hrtc模块，在package.json的开发依赖里引入hrtc ，如："hrtc": "./sdk/RtcSdk_Web_*.*.*.***.tar.gz"。在终端执行安装命令（版本号按实际替换）： npm install ，然后通过以下方式访问：

```
import HRTC from 'hrtc'
console.log(HRTC.VERSION)
```

----结束

8.3 SDK 使用

步骤1 检测浏览器是否兼容 SDK。具体接口详情请参见**checkSystemRequirements**。

```
async isBrowserSupport() {
  let check = false
  try {
    check = await HRTC.checkSystemRequirements()
    console.warn('browser isSupport: ' + check)
  } catch (error) {
    console.error(`check browser isSupport error: ${error.getCode()} - ${error.getMsg()}`)
    if (error.getCode() !== 90100025) {
      console.error(`browser Support part ability of RTC`)
      check = true
    }
  }
  return check
}
```

步骤2 创建客户端。具体接口详情请参见**createClient**。

```
let config = { appId,domain,countryCode }
let client = HRTC.createClient(config)
```

- **domain**：string[128]类型，服务器域名。该参数在SDK 1.0+版本中必填，SDK 2.0+版本中非必填。
- **appId**：string[128]类型，必填。应用ID，只有App ID相同的应用程序才能进入同一个房间进行互通。
- **countryCode**：string[2]类型，可选。国家码，如：CN表示中国大陆，US表示美国，HK表示中国香港。countryCode值的填写具体请参见**国家码对照表**。

说明

domain和**appId**请参考**应用管理**进行获取。

步骤3 加入房间。具体接口详情请参见**join**。

```
let option = { userId: userId, userName: userName, signature: signature, ctime: ctime, role: role }
async joinRoom() {
  try{
    await client.join(roomId, option)
    console.log('join room success')
  } catch(error){
    console.log('join room fail',error)
  }
}
```

- **userId**：必选，string[64] 类型，本端用户唯一标识。
- **userName**：可选，string[256]类型，用户昵称，该昵称为UTF-8编码。
- **signature**：必选，string[512]类型，鉴权签名字串，应用开发者需要向远端服务器获取鉴权签名。

  说明

  远端服务器需要您自行部署，具体请参见**接入鉴权**。
- **ctime**：必选，string类型，签名UTC时间戳，单位秒。
- **role**：必选，number类型，用户角色，可以标识媒体方向，取值如下：
  - 0：joiner（发布并观看）。
  - 2：player（只观看不发布）。

- **roomId**：必选，string[64]类型，房间ID，房间唯一标识。

加入房间成功后，对端会收到“peer-join”事件。

步骤4 创建本地流并发布。具体接口详情请参见**createStream**、**initialize**、**addResolution**、**publish**、**play**。

```
let stream = HRTC.createStream({ audio:true,microphoneId:xxx,video:true,cameraId:xxx })


stream.initialize().then(() => {
  stream.addResolution('90p_1')  //可选，如果要开启双流可以添加另外一个分辨率的视频

  stream.play(elementId,{ muted:true })  //播放本地流
  client.publish(stream)
})
```

- **audio**：可选，boolean类型，指定是否采集主流的音频。默认值为false。
- **video**：可选，boolean类型，指定是否采集主流的视频，主流即摄像头的流。默认值为false。
- **microphoneId**：可选，string类型，在audio为true的时候有效，表示采集音频的源麦克风设备Id。如果不传，系统自动设置默认值。
- **cameraId**：可选，string类型，在video为true的时候有效，表示采集视频的摄像头设备Id。如果不传，系统自动设置默认值。

步骤5 当收到服务器发送的“stream-added”事件通知时，可以订阅远端媒体。具体接口详情请参见**stream-added**、**subscribe**、**getStreamInfo**。

```
client.on('stream-added', (event) => {
  const stream = event.stream
  client.subscribe(stream,{ video:true, audio:true })
})
```

说明

```
如果双流场景：
client.on('stream-added', (event) => {
  const stream = event.stream
  const streamInfo = stream.getStreamInfo()  //获取流的分辨率等信息
  const resolutionIds = streamInfo.videoProfiles.map((profile) => profile.resolutionId)  // App 根据自己的业务场景，选择分辨率
  client.subscribe(stream,{ video:true, audio:true, resolutionIds:resolutionIds })  // 订阅音频以及所选择的分辨率的视频
  })
```

订阅完成之后，本端会收到“stream-subscribed”事件通知，可设置对端窗口，播放对端音视频。具体接口详情请参见**stream-subscribed**、**play**。

```
client.on('stream-subscribed', (event) => {
   const stream = event.stream
   stream.play(elementId, { objectFit: 'contain', muted: true, resolutionId: resolutionId })
})
```

- **elementId**：HTML 标签ID。
- 播放选项参数：
  - **objectFit**：可选，string类型，取值包括contain、cover和fill，默认值：主流：cover，辅流：contain。
  - **muted**：可选，boolean类型，true表示静音，false表示不静音，默认值为false。
  - **resolutionId**: 可选，string类型，指定要播放的分辨率的视频。默认为分辨率最高的视频。

若是不想观看对端，则可取消订阅对端音视频。具体接口详情请参见**unsubscribe**。


```
client.unsubscribe(stream)
```

步骤6 当远端离开房间，本端会收到“peer-leave”事件通知，清理远端用户的资源。具体接口详情请参见**peer-leave**。

```
client.on('peer-leave', (event) => {
  // just do something…
})
```

event.userId：对端用户标识，通过监听“peer-leave”事件获得。

远端用户退出，本端同时会收到“stream-removed”事件通知，可在事件处理函数中，关掉视频窗口。具体接口详情请参见**stream-removed**。

```
client.on('stream-removed', (event) => {
  event.stream.close()
})
```

通过stream对象调用**close**方法，该方法会移除之前用“play”创建的 video 标签元素并关闭摄像头、麦克风。

步骤7 本端离开房间。具体接口详情请参见**leave**。

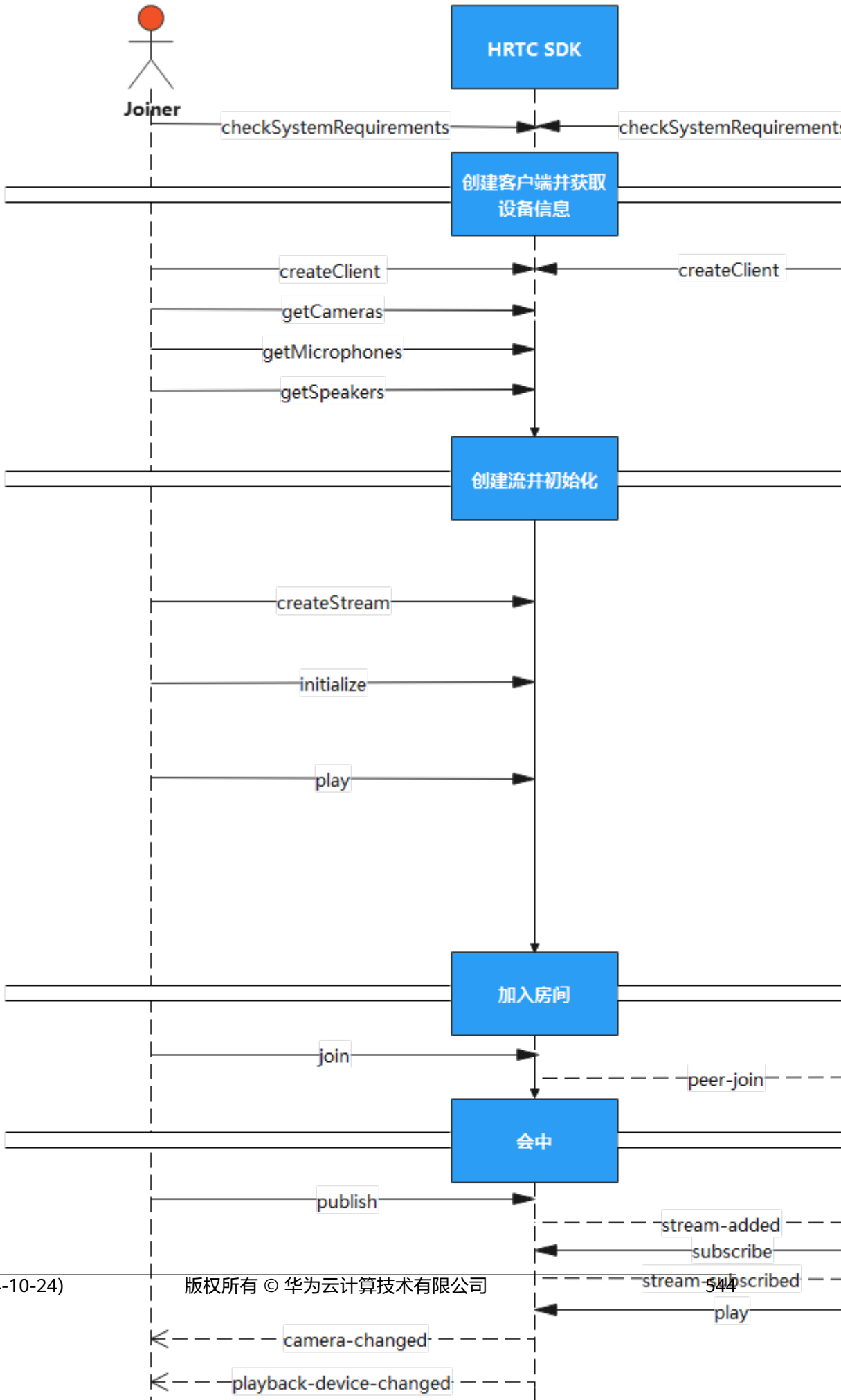
```
client.leave()
```

当音视频通话结束时，调用此接口离开房间。

至此，音视频通话基本流程可以成功运行。

----结束

8.4 基本使用逻辑



1. 创建新的项目工程，导入 SDK 后，需要创建客户端并获取本地音视频设备信息。
2. 创建本地流并初始化。
3. 当用户加入房间后，将通过回调的方式通知房间内的其他用户，收到用户加入的回调后，可以对音视频流进行订阅、取消订阅等其它操作。
4. 在会中，也可以对本地录音或播放设备等进行配置。
5. 用户离开房间后，房间内其他用户会收到该用户离开的回调信息，离开房间后，需销毁对应资源。

说明

在**时序图**中，单击相应接口名称可快速跳转到相应接口位置查看其使用方法。

8.5 接口参考

8.5.1 主入口（HRTC）

本章节介绍了Web SDK的HRTC接口详情。

表 8-3 主入口接口

接口	描述
checkSystemRequirements	检测浏览器是否兼容SparkRTC Web SDK。 **须知** 如果Web SDK版本在2.0.2到2.0.9.300之间，需要在2023年10月29日前更新至2.0.9.301及之后的版本，否则会导致checkSystemRequirements接口不可用。影响范围为微信浏览器。
VERSION	获取SparkRTC Web SDK版本。
getDevices	获取媒体输入输出设备列表。
getCameras	获取摄像头设备列表。
getMicrophones	获取麦克风设备列表。
getSpeakers	获取扬声器设备列表。
isScreenShareSupported	检查是否支持屏幕共享。
createClient	创建一个实时音视频通话的客户端对象。一个Client对应于一个房间。
createStream	创建一个本地流对象。流分为主流和辅流。主流指从摄像头和麦克风采集的音视频流，辅流是从共享屏幕采集的音视频流。
setLogLevel	设置日志级别。

checkSystemRequirements

须知

如果Web SDK版本在2.0.2到2.0.9.300之间，需要在2023年10月29日前更新至2.0.9.301及之后的版本，否则会导致checkSystemRequirements接口不可用。影响范围为微信浏览器。

```
(static) checkSystemRequirements(strictCheckWechatBrowser
: boolean): Promise<boolean>
```

【 功能说明 】

检测浏览器是否兼容SparkRTC Web SDK。

【 请求参数 】

strictCheckBrowser：可选，boolean类型，默认值为true 。true表示使用白名单放通支持的移动端浏览器（白名单由华为侧统一配置管理），false表示不使用白名单，该参数为2.0.2版本新增。

【 返回参数 】

Promise<boolean>：返回一个Promise对象，true表示浏览器兼容SparkRTC Web SDK，如果不兼容，则返回对应Error异常。

VERSION

```
VERSION
```

【 功能说明 】

获取 SparkRTC Web SDK 版本。

【 请求参数 】

无

【 返回参数 】

string：SDK当前版本号。

getDevices

```
(static) getDevices(): Promise<MediaDeviceInfo[]>
```

【 功能说明 】

获取媒体输入输出设备列表。在用户未授权摄像头或麦克风访问权限之前，“label”及“deviceId”可能为空。因此，建议在用户授权访问后，再调用此接口获取设备列表。

授权浏览器的摄像头/麦克风访问权限的方法，请参见**授权浏览器摄像头/麦克风访问权限的方法**。

【 请求参数 】

无

【 返回参数 】

Promise<MediaDeviceInfo[]>：媒体输入输出设备列表。**MediaDeviceInfo**为Web API基本接口。

getCameras

```
(static) getCameras(): Promise<MediaDeviceInfo[]>
```

【 功能说明 】

获取摄像头设备列表。在用户未授权摄像头访问权限之前，“label”及“deviceId”可能为空。因此建议在用户授权访问后，再调用此接口获取设备列表。

授权浏览器的摄像头/麦克风访问权限的方法，请参见**授权浏览器摄像头/麦克风访问权限的方法**。

【 请求参数 】

无

【 返回参数 】

Promise<MediaDeviceInfo[]>：摄像头设备列表。**MediaDeviceInfo** 为Web API基本接口。

getMicrophones

```
(static) getMicrophones(): Promise<MediaDeviceInfo[]>
```

【 功能说明 】

获取麦克风设备列表。在用户未授权麦克风访问权限之前，“label”及“deviceId”可能为空。因此建议在用户授权访问后，再调用此接口获取设备列表。

授权浏览器的摄像头/麦克风访问权限的方法，请参见**授权浏览器摄像头/麦克风访问权限的方法**。

【 请求参数 】

无

【 返回参数 】

Promise<MediaDeviceInfo[]>：麦克风设备列表。**MediaDeviceInfo**为Web API基本接口。

getSpeakers

```
(static) getSpeakers(): Promise<MediaDeviceInfo[]>
```

【 功能说明 】

获取扬声器设备列表。

【 请求参数 】

无

【 返回参数 】

Promise<MediaDeviceInfo[]>：扬声器设备列表。**MediaDeviceInfo**为Web API基本接口。

isScreenShareSupported

```
(static) isScreenShareSupported(): boolean
```

【 功能说明 】

检查浏览器是否支持屏幕共享。

【 请求参数 】

无

【 返回参数 】

boolean：true表示支持，false表示不支持。

createClient

```
(static)createClient(config: ClientConfig): Client
```

【 功能说明 】

创建一个实时音视频通话的客户端对象。一个客户端对象只能同时加入一个房间。可以创建多个客户端对象同时加入多个房间。

【 请求参数 】

config：必选，ClientConfig类型, 客户端对象配置信息。

ClientConfig定义为：{

- appId：必选，string[128]类型，应用ID，只有App ID相同的应用程序才能进入同一个房间进行互动。应用的appId请参考**应用管理**进行获取。
- domain：可选，string[128]类型，服务器的域名。需要与注册到SparkRTC平台的合法企业域名保持一致，该参数在SDK 1.0+版本中必填，SDK 2.0+版本中非必填。
- countryCode: 可选，string[2]类型，需要满足**ISO 3166-1**的2位字母的国家码要求。表示业务接入点的国家码，SDK会根据设置将业务接入到对应区域的服务，如：CN表示中国大陆，US表示美国，HK表示中国香港。countryCode值的填写具体请参见**国家码对照表**。该参数为2.0.3版本新增，且为必选参数，从2.0.7版本开始，修改为可选参数。

}

【 返回参数 】

Client：客户端对象。

createStream

```
(static) createStream(config: StreamConfig): Stream
```

【 功能说明 】

创建本地流对象。

【 请求参数 】

config：必选，StreamConfig类型，指定创建流的参数。

StreamConfig 定义为：{

- screen：可选，boolean类型，如果为true表示该流对象采集的是辅流音视频。辅流即共享屏幕的流。默认值为false，即该流对象采集的是主流音视频。
- video：可选，boolean类型，指定是否采集主流的视频，主流即摄像头的流。默认值为false。
- audio：可选，boolean类型，指定是否采集主流的音频。默认值为false。screen为false的时候该参数有效。
- microphoneId：可选，string类型，在audio为true的时候有效，表示采集音频的源麦克风设备Id。如果不传，系统自动设置默认值。
- cameraId：可选，string类型，在video为true的时候有效，表示采集视频的摄像头设备Id。如果不传，系统自动设置默认值。
- facingMode：可选，string类型，在video为true的时候有效，user表示前置摄像头，environment表示后置摄像头。
- screenAudio：可选，boolean类型，是否包含屏幕共享背景音。默认值为false。该功能仅支持Windows平台Chrome浏览器 74及以上版本。该字段为1.4.0版本新增。
- audioSource：可选，MediaStreamTrack类型，表示输入音轨对象。通过指定Track设置音频。**MediaStreamTrack**为Web API基本接口。
- videoSource：可选，MediaStreamTrack类型，表示输入视轨对象。通过指定Track设置主流视频。**MediaStreamTrack**为Web API基本接口。
- mirror：可选，boolean类型，表示主流的本地视频是否镜像。默认值为false。
- userId：可选，string类型，表示该流归属的userId。

}

【 返回参数 】

Stream：流对象。

⚠ 注意

- 在采集主流有两种方式：
  - 通过“audioSource”和“videoSource”设置音频和视频主流。这种模式下不支持大小流。
  - 通过“audio/microphoneId”和“video/cameraId/facingMode”设置音频和视频主流。
- 如果未指定任何的音频源和视频源，则创建的流对象不包含音频流和视频流，无法播放。
- 如果采集视频，同一个Stream对象不能同时采集主流和辅流。
- 如果需要包含屏幕共享背景音，需要设置screen和screenAudio均为true，该参数仅在1.4.0 及以上版本生效。

setLogLevel

```
(static) setLogLevel(level: LogLevel): void
```

【 功能说明 】

设置日志输出等级。默认输出info等级日志。

【请求参数】

level：必选，LogLevel类型，设置日志级别。

LogLevel表示日志级别，枚举取值如下：

- none：string类型，表示关闭SDK日志打印。
- error：string类型，表示开启SDK错误日志级别。
- warn：string类型，表示开启SDK警告日志级别。
- info：string类型，表示开启SDK信息日志级别。
- debug：string类型，表示开启SDK调试日志级别。

【返回参数】

无

8.5.2 客户端对象（Client）

本章节介绍了Web SDK的Client接口详情。

表 8-4 Client 接口

接口	描述
join	加入房间。调用该接口让用户加入指定房间，进行音频/视频通话。
leave	离开房间。用户结束通话后须调用该接口离开房间。
publish	加入房间后，发布本地流。
unpublish	取消发布本地流。
subscribe	订阅远端音视频媒体流。
unsubscribe	取消订阅远端音视频媒体。
batchSubscribe	批量订阅远端用户的视频媒体流。
switchRole	切换用户角色。
on	注册客户端对象事件回调接口。
off	取消注册客户端对象事件回调接口。
getConnectionState	获取客户端连接状态。
getTransportStats	获取当前网络传输状况统计数据。
getLocalAudioStats	获取本地音频统计数据。
getLocalVideoStats	获取本地视频统计数据。
getRemoteAudioStats	获取远端音频统计数据。

接口	描述
getRemoteVideoStats	获取远端视频统计数据。
enableTopThreeAudioMode	设置是否开启音频TopN模式。
setVolume4TopThree	设置音频TopN模式的音量值。该接口为1.4.0版本新增。
muteAudio4TopThree	开启/禁用音频TopN模式的所有音轨。该接口为1.4.0版本新增。
enableStreamStateDetection	开启/关闭视频码流状态探测功能。该接口为1.4.0版本新增。
changeUserName	修改用户昵称。该接口为1.5.0版本新增。
startLiveStreaming	启动直播流旁推到CDN。
updateLiveStreaming	更新旁路推流的合流参数。
stopLiveStreaming	停止直播流旁推到CDN。
setProxyServer	设置企业内部部署代理反向服务器。该接口为2.0.3版本新增。
setTurnServer	设置代理服务器配置信息。该接口为2.0.3版本新增。
enableRtcStats	设置是否开启RTC音视频流统计信息事件。该接口为2.0.3版本新增。
setNetworkBandwidth	设置媒体最大带宽。该接口为2.0.5版本新增。
renewSignature	更新签名。该接口为2.0.8版本新增。

join

```
async join(roomId: string, options: JoinConfig): Promise<void>
```

【功能说明】

加入房间。该接口让用户加入一个房间，进行音频或视频通话。

【请求参数】

- roomId：房间ID，房间唯一标识符。必选，string[64]类型。
- options：加入房间配置。必选，JoinConfig类型。
  JoinConfig 定义为：{
  - userId：用户标识，userId需要保证应用内唯一。必选，string[64]类型。
  - userName：用户昵称。可选，string[256]类型。
  - signature：签名，签名的具体生成方法请参见接入鉴权。必选，string[512]类型。

- ctime：必选，鉴权签名时间戳。string类型，UTC时间戳，单位秒。
- role：必选，number类型，用户角色，可以标识媒体方向。role的枚举值包括：
  - 0： 表示joiner，能够发送音视频和接收音视频。
  - 2： 表示player，只接收别人的音视频，不发送自己的音视频媒体。

【返回参数】

Promise<void>：返回一个Promise对象。

⚠ 注意

- roomId支持的字符包括：a-z、A-Z、0-9、连接符 '-'、下划线 '_'。
- userId支持的字符包括：a-z、A-Z、0-9、连接符 '-'、下划线 '_'。

leave

```
async leave(): Promise<void>
```

【功能说明】

离开房间，用户结束通话后须调用该接口离开房间。

【请求参数】

无

【返回参数】

Promise<void>：返回一个Promise对象。

publish

```
async publish(stream: Stream, option?: PublishOption): Promise<void>
```

【功能说明】

加入房间并创建本地流后，可以调用该接口发布本地流。只有角色为“joiner”才能发布本地流。

【请求参数】

- stream：必选，Stream类型，本地流对象。
- option：可选，PublishOption类型，表示是否主动发布视频流，如果不传，则默认不主动发布视频流。

PublishOption类型定义如下：

{

autoPushVideo：可选，boolean类型，表示是否主动发布视频流，默认为false。

}

【返回参数】

Promise<void>：返回一个Promise对象。

unpublish

```
async unpublish(stream: Stream): Promise<void>
```

【功能说明】

取消发布本地流。

【请求参数】

stream：必选，Stream类型，本地流对象。

【返回参数】

Promise<void>：返回一个Promise对象。

subscribe

```
async subscribe(stream: RemoteStream, option?: SubscribeOption): Promise<void>
```

【功能说明】

订阅远端音视频媒体流。订阅远端媒体流成功后，会收到Client.on('**stream-subscribed**') 事件通知。然后可以对流进行播放。

【请求参数】

- stream：必选，RemoteStream类型，远端流对象，通过**stream-added**事件获得。
- option：可选，SubscribeOption类型，表示订阅视频或音频的选项，如果不传，则订阅音频和分辨率最高的视频。

SubscribeOption 类型定义如下：

{

- video：可选，boolean 类型，表示是否订阅视频，默认为 false。
- audio：可选，boolean 类型，表示是否订阅音频，默认为 false。
- resolutionIds：可选。string[]类型，video为true的时候有效。表示要订阅的分辨率Id的视频。resolutionId可通过接口**getStreamInfo** 获取。如果不传resolutionIds，则默认订阅分辨率最高的一个视频。该参数为1.8.0 版本新增。

}

【返回参数】

Promise<void>：返回一个Promise对象。

⚠ 注意

SubscribeOption中“video”和“audio”不能同时为“false”。

unsubscribe

```
async unsubscribe(stream: Stream,option?: SubscribeOption): Promise<void>
```

【功能说明】

取消订阅远端音视频媒体流。

【请求参数】

- stream：必选，RemoteStream类型，远端流对象，通过**stream-added**事件获得。
- option：可选。表示取消订阅的选项，如果不传，则取消订阅音频和所有分辨率的视频，为SubscribeOption类型。

  SubscribeOption定义如下：{
  - video：可选，boolean类型，表示是否取消订阅视频，默认为false。
  - audio：可选，boolean类型，表示是否取消订阅音频，默认为false。
  - resolutionIds：可选，string[]类型，在video为true的时候有效，表示取消订阅的分辨率Id的视频。resolutionId通过接口**getStreamInfo** 获取。如果resolutionIds不传，则取消订阅所有分辨率的视频。

  }

【返回参数】

Promise<void>：返回一个Promise对象。

batchSubscribe

```
async batchSubscribe(subscribeInfos: SubscribeParam[]): Promise<void>
```

【功能说明】

批量订阅多个用户的视频。订阅远端媒体流成功后，对每个远端用户都会收到Client.on('**stream-subscribed**')事件通知。然后可以对订阅成功的流进行播放。

【请求参数】

subscribeInfos：必选，SubscribeParam[]类型，表示要订阅的全量远端用户的视频信息。

SubscribeParam类型定义如下：{

- userId：必选，string类型，表示要订阅的用户ID。
- resolutionIds：可选。string[]类型，表示要订阅的用户的视频的分辨率。如果不传resolutionIds，则默认订阅分辨率最高的一个视频。
- minResolution：可选，ResolutionType类型，表示如果开启视频分辨率自适应升降档后，最小的分辨率档位。ResolutionType包括的字符串枚举值包括FHD，HD，SD，LD。

}

【返回参数】

Promise<void>：返回一个Promise对象。

⚠ 注意

- 该接口订阅远端用户的视频，不订阅远端用户的音频。
- 不同于**subscribe** 接口的增量订阅方式，该接口订阅的入参是要订阅的全量用户列表。

switchRole

```
async switchRole(role: number,authorization: Authorization): Promise<void>
```

【功能说明】

切换用户角色，本方法用于加入房间成功之后修改角色。

【请求参数】

- role：必选，number类型。
  - 0：表示joiner，能够发送音视频和接受音视频。
  - 2：表示player，只接受别人的音视频，不发送自己的音视频媒体。
- authorization：必选，类型为Authorization，表示鉴权信息。该字段为2.0.0版本新增。Authorization的属性包括：
  - signature：必选，string[512]类型，鉴权签名字串，具体生成方法请参见**接入鉴权**。
  - ctime：必选，鉴权签名时间戳。string类型，UTC时间戳，单位秒。

【返回参数】

Promise<void>：返回一个Promise对象。

on

```
on(event: string, handler: function): void
```

【功能说明】

注册客户端对象事件回调接口。

【请求参数】

- event：必选，string类型，事件名称。具体请参见**ClientEvent**。
- handler：必选，function类型，事件处理方法。

【返回参数】

无

off

```
off(event: string, handler: function): void
```

【功能说明】

取消注册客户端对象事件回调接口。

【请求参数】

- event：必选，string类型，事件名称。具体请参见**ClientEvent**。
- handler：必选，function类型，事件处理方法。

【返回参数】

无

getConnectionState

```
getConnectionState(): ConnectionState
```

【功能说明】

获取客户端连接状态。

【请求参数】

无

【返回参数】

ConnectionState：webSocket的连接状态，string类型，取值如下：

- CONNECTING：连接建立中。
- CONNECTED：连接已建立。
- RECONNECTING：重新连接中。
- DISCONNECTED：连接已断开。

getTransportStats

```
getTransportStats(): Promise<TransportStats>
```

【功能说明】

获取当前网络传输状况统计数据，该方法需要**publish**后调用。双流场景，默认获取主流最高分辨率视频的统计值。

【请求参数】

无

【返回参数】

返回参数为TransportStats类型，TransportStats当前支持的属性包括：{

- bytesSent：number类型，表示已发送字节数。
- bytesReceived：number类型，表示已接收字节数。
- sendBitrate：number类型，表示当前出流码率，单位kbps。
- recvBitrate：number类型，表示当前入流码率，单位kbps。
- rtt：number类型，表示SDK到边缘服务器的RTT (Round-Trip Time)，单位毫秒。

  }

getLocalAudioStats

```
getLocalAudioStats(): Promise<Map<string, LocalAudioStats>>
```

【功能说明】

获取本地已发送的音频统计数据。

【请求参数】

无

【返回参数】

- string：表示userId。
- LocalAudioStats：表示本地音频统计指标，包含如下属性：{

- bytesSent：number类型。表示已发送字节数。
- packetsSent：number类型。表示已发送包数。

  }

getLocalVideoStats

```
getLocalVideoStats(): Promise<Map<string, AllLocalVideoStats>>
```

【功能说明】

获取本地已发送的视频统计数据。

【请求参数】

无

【返回参数】

- string：表示userId。
- AllLocalVideoStats：本地视频统计指标。包含如下属性：{
  - mainStream：LocalVideoStats[]类型，本地主流视频的统计数据。
  - subStream：LocalVideoStats类型，本地辅流视频的统计数据。

  }

  LocalVideoStats包含如下属性：{
  - bytesSent：number类型，已发送字节数。
  - packetsSent：number类型，已发送包数。
  - framesEncoded：number类型，已编码帧数。
  - framesSent：number类型，已发送帧数。
  - frameWidth：number类型，视频宽度。
  - frameHeight：number类型，视频高度。

  }

⚠ 注意

getLocalVideoStats和getLocalAudioStats接口仅在本地流发布且被远端订阅后才可调用。

getRemoteAudioStats

```
getRemoteAudioStats(): Promise<Map<string, RemoteAudioStats>>
```

【功能说明】

获取远端音频统计数据。

【请求参数】

无

【返回参数】

- string：表示userId。

- RemoteAudioStats：远端音频统计指标。包含如下属性：
  - bytesReceived：number类型，已接收字节数。
  - packetsReceived：number类型，已接收包数。
  - packetsLost：number类型，丢包数。

getRemoteVideoStats

```
getRemoteVideoStats(): Promise<Map<string, AllRemoteVideoStats>>
```

【功能说明】

获取远端视频统计数据。

【请求参数】

无

【返回参数】

- string：表示userId。
- AllRemoteVideoStats：远端视频统计指标。包含如下属性：{
  - mainStream：RemoteVideoStats[]类型，远端主流（包括大小流）的统计数据。
  - subStream：RemoteVideoStats类型，远端辅流的统计数据。

  }

  RemoteVideoStats 包含如下属性：{
  - bytesReceived：number类型，已接收字节数。
  - packetsReceived：number类型，已接收包数。
  - packetsLost：number类型，丢包数。
  - framesDecoded：number类型，已解码帧数。
  - frameWidth：number类型，视频宽度。
  - frameHeight：number类型，视频高度。

  }

⚠ 注意

Firefox浏览器上，无法获取到远端视频的分辨率数据。

enableTopThreeAudioMode

```
enableTopThreeAudioMode(enable: boolean): boolean
```

【功能说明】

设置入会前是否开启音频TopN 模式（最大三方模式）。

【请求参数】

enable：必选，boolean类型，true表示开启音频TopN 模式，false表示不开启。默认为false。

【返回参数】

boolean类型，true表示开启成功，false表示开启失败。

注意

该接口需在加入房间前设置，1.2.0版本新增。

setVolume4TopThree

```
setVolume4TopThree(volume: number): void
```

【功能说明】

开启音频TopN模式（最大三方模式）后，设置音频的音量值。

【请求参数】

volume：必选，number类型，取值范围为[0,100]，音频的音量值。

【返回参数】

无

注意

该接口需要在**enableTopThreeAudioMode**后设置，1.4.0版本新增。

muteAudio4TopThree

```
muteAudio4TopThree(enable: boolean): void
```

【功能说明】

开启音频TopN模式（最大三方模式）后，开启/禁用音频TopN模式的音轨。

【请求参数】

enable：必选，boolean类型，true表示禁用音频TopN模式的音轨，false表示开启音频TopN模式的音轨。默认为false。

【返回参数】

无

注意

该接口需要在**enableTopThreeAudioMode**后设置，1.4.0版本新增。

enableStreamStateDetection

```
async enableStreamStateDetection(enable: boolean, interval: number): Promise<void>
```

【功能说明】

开启/关闭视频码流状态探测功能，开启后可探测房间内任意其他已订阅用户是否处于视频无码流的状态，若有用户处于视频无码流状态，可收到**stream-interrupted**事件，若有用户视频码流恢复，可收到**stream-recovered**事件。

【 请求参数 】

- enable：必选，boolean类型，true表示开启视频码流状态探测，false表示关闭视频码流状态探测。默认为false。
- interval：必选，number类型，单位：秒 ，取值范围为[1,60]。视频无码流状态的判断时间。推荐设置为3秒。

【 返回参数 】

Promise<void>：返回一个Promise对象。

⚠ 注意

该接口需在加入房间后设置，1.4.0版本新增。

changeUserName

```
async changeUserName(userName: string): Promise<boolean>
```

【 功能说明 】

Joiner角色的用户修改用户昵称，修改成功后，房间内其他用户会收到**remote-user-name-changed**事件，而Player角色的用户修改后不会通知房间内的其他用户。

【 请求参数 】

userName：必选，string[256]类型，新的用户昵称。

【 返回参数 】

Promise<boolean>：返回一个Promise对象，true表示修改成功，false表示修改失败。

⚠ 注意

该接口需在加入房间后设置，1.5.0版本新增。

startLiveStreaming

```
async startLiveStreaming(taskId: string, urls: string[], publishConfig: PublishConfig, userConfigs: UserConfig[]): Promise<void>
```

【 功能说明 】

启动直播流旁推到CDN。

【 请求参数 】

- urls：必选，string[]类型，旁推的CDN Urls。
- taskId：必选， string类型，表示旁推任务Id。由APP生成和管理。

- publishConfig：必选，PublishConfig类型，表示旁推音视频配置。PublishConfig定义如下：{
  - width：必选，number类型，表示推流视频的总宽度，默认值360，单位为像素。
  - height：必选，number类型，表示推流视频的总高度，默认值640，单位为像素。
  - videoBitrate：必选，number类型，表示推流视频的码率，单位为Kbps，默认值为400 Kbps。
  - videoFramerate：必选，number类型，表示推流视频的帧率，单位为fps，取值范围为[0, 60]，默认值为15 fps。
  - videoGop：必选，number类型，表示旁路直播的输出视频的GOP，单位为帧。默认值为30帧。
  - audioSampleRate：必选，number类型，表示旁路直播的输出音频的采样率16000/32000/48000，默认值为32000。
  - audioBitrate：必选，number类型，用于旁路直播的输出音频的码率。单位为Kbps，默认值为48。
  - audioChannels：必选，number类型，用于旁路直播的输出音频的声道数，取值范围为[1, 5]，默认值为2。
  - template：可选，number类型，0表示悬浮，1表示九宫格，2表示屏幕分享，99表示自定义模板，默认值为0。

    }
- userConfigs：必选，UserConfig[]类型。 表示需要旁推的每个用户的配置。UserConfig定义如下：{
  - userId：必选，string类型，表示流用户Id。
  - audio：必选，boolean类型，表示是否旁推音频。
  - resolutionIds：必选，string[]类型，表示要旁推的主流或者辅流的视频分辨率ID。
  - layouts：可选，ResolutionLayout[]类型 ，在template为自定义模板有效，表示自定义用户流画面布局。ResolutionLayout定义如下：{
    - resolutionId：必选，string类型，表示主流或者辅流的视频分辨率ID。
    - subWidth：可选，number类型，表示画面在输出时的宽度，单位为像素值，默认值为36。
    - subHeight：可选，number类型，表示画面在输出时的高度，单位为像素值，默认值为64。
    - localX：可选，number类型，表示该画面在输出时的X偏移，单位为像素值，localX与subWidth之和不能超过混流输出的总宽度，默认值为0。
    - localY：可选，number类型，表示该画面在输出时的Y偏移，单位为像素值，localY与subHeight之和不能超过混流输出的总高度，默认值为0。
    - renderMode：可选，number类型，表示该画面在输出时的显示模式，0表示裁剪，1表示缩放。默认值为0。

- alpha：可选，number类型，表示该用户视频图像在输出视频图像中的透明度，取值范围为0~100。0表示该用户视频图像完全透明，100表示该用户视频图像完全不透明。默认值为100。
- subBackGroundColor：可选，number类型，表示混流-输出流背景色，取值是十进制整数。默认为黑色。
- zorder：可选，number类型，标志图层。

}
}

【返回参数】

Promise<void>：返回一个Promise对象。

⚠ 注意

- 该方法只有角色为joiner的用户才能使用该功能，2.0.1版本新增。
- 请确保在publish流成功之后调用本方法。

updateLiveStreaming

```
async updateLiveStreaming(taskId: string, urls: string[], publishConfig: PublishConfig, userConfigs: UserConfig[]): Promise<void>
```

【功能说明】

更新旁路推流。

在推流状态变更时，本地会触发Client.on(“**live-streaming-updated**”) 回调。

【请求参数】

和**startLiveStreaming**相同。

【返回参数】

Promise<void>：返回一个Promise对象。

⚠ 注意

该方法需在启动直播流旁推成功后调用，2.0.1版本新增。

stopLiveStreaming

```
async stopLiveStreaming(taskId: string): Promise<void>
```

【功能说明】

停止直播流旁推到CDN。该方法在2.0.1版本新增。

【请求参数】

taskId：必选，string类型，旁路推流的任务Id。

【返回参数】

Promise<void>：返回一个Promise对象。

enableRtcStats

```
async enableRtcStats(enable: boolean, interval: number): Promise<void>
```

【功能说明】

设置是否开启RTC音视频流统计信息事件。该接口为2.0.3版本新增。

【请求参数】

enable：必选，boolean类型，表示是否开启RTC音视频流统计信息事件，true表示开启，false表示不开启。

interval：必选，number类型，设置RTC音视频流统计信息事件间隔时间，单位毫秒。当enable为true时有效。

【返回参数】

Promise<void>：返回一个Promise对象。

setProxyServer

```
setProxyServer(server: string): void
```

【功能说明】

设置信令代理服务器。用于企业内部部署反向代理服务器（如 nginx ）的场景。该方法为2.0.3版本新增。

【请求参数】

server：必选，string类型，反向代理服务器的列表。一个代理服务器的格式，比如：http://ip:port / https://domain:port。

【返回参数】

无

⚠ 注意

setProxyServer和setTurnServer方法需在join之前调用。

setTurnServer

```
setTurnServer(turnServerConfig: TurnServerConfig): void
```

【功能说明】

设置代理服务器配置信息。用于企业内部部署反向代理服务器（如nginx）的场景。该方法为2.0.3版本新增。

【请求参数】

turnServerConfig：代理服务器配置信息。必选，TurnServerConfig类型。

TurnServerConfig 类型定义如下：{

- turnServers：反向代理服务器地址。必选，string[]类型。
- udpPort：UDP端口。可选，number类型。
- userName：反向代理服务器用户名。可选，string类型。
- credential：反向代理服务器密码。可选，string类型。

  }

【 返回参数 】

无

setNetworkBandwidth

```
setNetworkBandwidth(bandwidthParam: NetworkBandwidth): void
```

【 功能说明 】

设置媒体最大带宽。该方法为2.0.5版本新增。

【 请求参数 】

NetworkBandwidth类型定义如下：{

maxBandwidth：必选，number类型，媒体最大总带宽，取值范围为[3072, 51200]，单位为kbps。

}

【 返回参数 】

无

renewSignature

```
renewSignature(ctime: string, signature: string): boolean
```

【 功能说明 】

更新签名。

【 请求参数 】

- ctime：签名鉴权的过期时间，是系统当前UTC时间（unix时间戳）加上鉴权过期时间（推荐2小时，最长需要小于12小时）。单位：秒。必选，string类型。
- signature：签名，签名的具体生成方法请参见**接入鉴权**。必选，string类型，string[512]类型。

【 返回参数 】

boolean：返回一个boolean值，说明更新签名是否成功。

⚠ 注意

该接口2.0.8版本新增。

8.5.3 客户端事件通知（ClientEvent）

本章节介绍了Web SDK的ClientEvent事件。

表 8-5 ClientEvent 事件

接口	描述
peer-join	远端用户进入房间事件。
peer-leave	远端用户退出房间事件。
stream-added	远端流添加事件。
stream-removed	远端流删除事件。
stream-updated	远端流更新事件。
stream-subscribed	远端流订阅成功事件。
client-banned	用户被踢，离开房间事件。
kick-room	解散房间
Error	客户端出现错误事件。
connection-state-changed	Client连接状态变更事件。
mute-audio	远端流禁用音频事件。
unmute-audio	远端流启用音频事件。
mute-video	远端流禁用视频事件。
unmute-video	远端流启用视频事件。
log-upload-result	日志上传结果事件。
signature-expired	签名过期事件。该事件为2.0.8版本新增。
camera-changed	摄像头设备变更事件。
recording-device-changed	录音设备变更事件。
playback-device-changed	播放设备变更事件。
network-quality	网络上下行质量报告事件。
stream-interrupted	远端流处于无码流状态事件。该事件为1.4.0版本新增。
stream-recovered	远端流码流恢复事件。该事件为1.4.0版本新增。
volume-indicator	音频TopN（即最大三方）模式下，当前音量最大的用户提示事件。该事件为1.5.0版本新增。
remote-user-name-changed	远端用户昵称变更事件。该事件为1.5.0版本新增。
live-streaming-updated	更新旁路推流的事件。该事件为2.0.0版本新增。
rtc-stats	音视频流数据统计事件。该事件为2.0.3版本新增。

⚠ 注意

事件注册监听应在业务结束时取消注册，否则注册监听事件累积会有内存泄漏风险。

peer-join

【事件说明】

远端用户加入房间事件，远端用户加入房间后会收到该事件通知。

【回调参数】

peerJoinEvent：必选，PeerJoin类型，用户信息。

PeerJoin定义为：{

- userId：必选，string[64]类型，用户标识。
- userName：可选，string[256]类型，用户昵称。

}

peer-leave

【事件说明】

远端用户离开房间事件，远端用户离开房间后会收到该事件通知。

【回调参数】

peerLeaveEvent：必选，PeerLeaveInfo 类型，用户离开信息。

PeerLeaveInfo 定义为：{

- userId：必选，string[64]类型，用户标识。
- userName：可选，string[256]类型，用户昵称。
- reason：可选，HRTCLeaveReason类型。

}

HRTCLeaveReason定义为：{

- code：number类型, 离开原因枚举，取值如**表2 离开房间原因**所示。
- msg：string类型，原因描述。

}

表 8-6 离开房间原因

枚举值	描述
0	用户主动离开
1	服务器异常
2	sfu服务故障

枚举值	描述
3	服务不可达503
4	内部错误
5	被踢出房间
6	签名过期
7	重连超时
8	网络检测，UI不需要关注该错误码，不对外体现
9	用户移除
10	房间解散

stream-added

【事件说明】

远端流添加事件，当远端用户发流成功后会收到该事件通知。

【回调参数】

stream：必选，RemoteStream类型，远端流对象。

stream-removed

【事件说明】

远端流移除事件，当远端用户取消发流或者原来已经发布的远端用户退出房间后会收到该事件通知。

【回调参数】

stream：必选，RemoteStream类型，远端流对象。

stream-updated

【事件说明】

远端流更新事件，当远端用户的流发生变化，例如添加或者移除音视频轨，视频轨规格变化时会收到该事件通知。

【回调参数】

stream：必选，RemoteStream类型，远端流对象。

stream-subscribed

【事件说明】

远端流订阅成功事件，当订阅远端流成功后会收到该事件通知。

【回调参数】

stream：必选，RemoteStream类型，远端流对象。

client-banned

【事件说明】

用户被踢下线事件，当用户以相同的userId 在其他Client加入相同的房间后，被踢的Client会收到该事件通知。

【回调参数】

clientBannedEvent：必选，ClientBanInfo类型，

ClientBanInfo定义为：{

- userId：必选，string[64]类型，被踢用户标识。
- reason：必选，string类型，原因描述。

}

kick-room

【事件说明】

解散房间，由调用的服务端接口触发。

【回调参数】

roomId：必选，string[64]类型，房间Id。

Error

【事件说明】

客户端错误事件，当出现不可恢复的错误后，Client会上报该事件通知。

【回调参数】

errorInfo：必选，ErrorInfo类型，错误信息。

ErrorInfo 定义为：{

- errorCode：必选，string类型，错误码。
- errorMsg：必选，string类型，错误描述。

}

connection-state-changed

【事件说明】

Client连接状态变更事件，Client连接状态变更会收到该事件通知。

【回调参数】

- ConnectionStateInfoEvent：{
  - prevState：必选，ConnectionState类型，变更前状态。
  - curState：必选，ConnectionState类型，变更后状态。

}

连接状态ConnectionState的取值如下所示：

- CONNECTING：连接建立中。
- CONNECTED：连接已连接。
- RECONNECTING：重新连接中。
- DISCONNECTED：连接已断开。

mute-audio

【事件说明】

远端用户禁用音频通知事件，当远端用户禁用音频后，流接收端会收到该事件通知。

【回调参数】

mediaStatus：必选，MediaStatusNotifyInfo类型。

MediaStatusNotifyInfo包含如下属性：

- roomId：必选，string[64]类型。
- userId：必选，string[64]类型。
- status：必选，MediaStatusAction类型。MediaStatusAction枚举值定义如下所示：
  - 1：媒资状态可用。
  - 2：媒资状态不可用。
- reason：必选，MediaStatusReason类型。MediaStatusReason枚举值定义如下所示：
  - 0：媒体离线。
  - 1：媒体静音。
  - 2：媒体不静音。

unmute-audio

【事件说明】

远端用户启用音频通知事件，当远端用户启用音频后，流接收端会收到该事件通知。

【回调参数】

mediaStatus：必选，**MediaStatusNotifyInfo**类型。

mute-video

【事件说明】

远端用户禁用视频通知事件，当远端用户禁用视频后，流接收端会收到该事件通知。

【回调参数】

mediaStatus：必选，**MediaStatusNotifyInfo**类型。

unmute-video

【事件说明】

远端用户启用视频通知事件，当远端用户启用视频后，流接收端会收到该事件通知。

【回调参数】

mediaStatus：必选，**MediaStatusNotifyInfo**类型。

log-upload-result

【事件说明】

日志上传结果事件。

【回调参数】

status：必选，number类型，日志上传结果，200表示成功，其余表示失败。

signature-expired

【事件说明】

签名过期事件。

【回调参数】

errorInfo：错误信息。必选，ErrorInfo类型。

ErrorInfo 定义为：{

- errorCode：错误码。必选，string类型。
- errorMsg：必选，string类型。取值如下面的取值示例中所示。

}

取值示例，如下所示：

- 签名过期：{
  errorCode: '90100030'
  errorMsg: 'signature expired'
  }
- 签名无效：{
  errorCode: '90100031'
  errorMsg: 'signature invalid'
  }

注意

监听到签名过期事件后可通过错误码区分是签名无效还是签名过期，签名过期后可通过**renewSignature**接口更新签名。

camera-changed

【事件说明】

本地摄像头设备变更事件，当插、拔本地摄像头时触发。

【回调参数】

DeviceChangedEvent：必选，DeviceChangedInfo类型，设备变更详情。

DeviceChangedInfo 定义为：{

- deviceId：必选，string类型，设备deviceId。
- state：必选，DeviceChangeMode类型，DeviceChangeMode枚举值定义如下：
  - ADD：设备添加。
  - REMOVE：设备移除。

}

⚠ 注意

视频采集设备，拔插后需要应用层进行相应的处理，如：拔除摄像头的时候是否切换其他视频采集设备重新采集；插入摄像头的时候是否使用新插入设备进行重新采集。

recording-device-changed

【事件说明】

本地录音设备变更事件，当本地录音设备变更时触发。

【回调参数】

DeviceChangedEvent：必选，DeviceChangedInfo类型，设备变更详情。
DeviceChangedInfo定义参考**camera-changed**中的说明。

⚠ 注意

麦克风采集设备，拔插后需要应用层进行相应的处理，如：拔除麦克风的时候是否切换其他麦克风采集设备重新采集；插入麦克风的时候是否使用新插入设备进行重新采集。

playback-device-changed

【事件说明】

本地音频播放设备变更事件，当本地音频播放设备变更时触发。

【回调参数】

DeviceChangedEvent：必选，DeviceChangedInfo类型，设备变更详情。
DeviceChangedInfo定义参考**camera-changed**中的说明。

network-quality

【事件说明】

网络上下行质量报告事件，用户加入房间后，SDK在网络质量变化的时候会触发一次该事件，报告用户的本地网络上下行质量情况。

【回调参数】

NetworkQualityEvent：必选， NetworkQualityInfo类型，网络上下行质量详情。

NetworkQualityInfo定义为：{

- uplinkNetworkQuality：必选，number类型，上行网络质量。枚举值如下：
  - 0：质量未知。
  - 1：质量极好。
  - 2：质量较好。
  - 3：质量一般。
  - 4：质量差。
  - 5：质量极差。
  - 6：网络连接断开。
- downlinkNetworkQuality：必选，number类型，下行网络质量。枚举值如下：
  - 0：质量未知。
  - 1：质量极好。
  - 2：质量较好。
  - 3：质量一般。
  - 4：质量差。
  - 5：质量极差。
  - 6：网络连接断开。

}

stream-interrupted

【事件说明】

远端流的码流中断事件。中断表示在 **enableStreamStateDetection** 接口参数 interval 设置的统计周期内未接收到有效视频帧。该事件为 1.4.0 版本新增。

【回调参数】

streamInterruptedEvent：必选，UserList[]类型，已订阅且无视频码流的用户列表。

UserList定义为：{

- userId：必选，string类型，用户Id。
- isScreen：必选，boolean类型，true表示对应用户的辅流，false表示对应用户的主流。

}

stream-recovered

【事件说明】

远端流的码流恢复事件。

【回调参数】

streamRecoveredEvent：必选，UserList[]类型，已订阅且视频码流恢复的用户列表。UserList参考**stream-interrupted**中的定义。该事件为1.4.0版本新增。

volume-indicator

【事件说明】

音频TopN模式下，房间中当前音量最大的用户提示事件。

【回调参数】

userVolumeInfos：必选， UserVolumeInfo[]类型。

UserVolumeInfo定义为：{

- user_id：必选，string类型，用户Id。
- volume：可选，number类型，取值范围为[0,100]。

}

⚠ 注意

该事件仅在音频TopN模式下生效，1.5.0版本新增。

remote-user-name-changed

【事件说明】

远端用户昵称变更事件。该事件为1.5.0版本新增。

【回调参数】

userNameChangedEvent：必选， UserNameInfo类型。

UserNameInfo定义为：{

- roomId：必选，string[64]类型，房间Id。
- userId：必选，string[64]类型，用户Id。
- userName：必选， string[256]类型，修改后的用户昵称。

}

live-streaming-updated

【事件说明】

旁路推流状态更新事件。该事件为2.0.1版本新增。

【回调参数】

urlStatus：UrlStatus[]类型，所有CDN推流状态更新。UrlStatus定义如下：

- url：必选，string类型，表示CDN推流URL

- status：必选，number类型，表示当前推流状态。
  - 0：初始化。
  - 1：链接正常且有流。
  - 2：链接正常但无流。
  - 3：异常重试。
  - 4：处理失败。
- errorCode：可选，number类型，表示详细的失败原因，支持的枚举值如下：
  - 0： 正常。
  - 1： 内部错误。
  - 2： 地址解析失败。
  - 3： 连接失败。
  - 4： RTMP握手失败。
  - 5： 内存错误。
  - 6： 参数错误。
  - 7： 重试失败。
  - 8： 响应超时失败。

rtc-stats

【事件说明】

音视频流数据统计事件。该事件为2.0.3版本新增。

【回调参数】

rtcStatsInfo：必选， rtcStatsInfo[]类型。

rtcStatsInfo定义为：{

- userName：必选，string类型，用户昵称。
- isRemote：必选，boolean类型，是否为远端流，true表示远端流，false表示本地流。
- streamType：必选，ContentType类型，流类型。ContentType类型的枚举值如下：{
  - main：string类型，表示主流。
  - middle1：string类型，表示主流，middle1~middle4码流依次降低。
  - middle2：string类型，表示主流，middle1~middle4码流依次降低。
  - middle3：string类型，表示主流，middle1~middle4码流依次降低。
  - middle4：string类型，表示主流，middle1~middle4码流依次降低。
  - slides：string类型，表示主流视频分辨率最小的流。
  - desktop：string类型，表示共享流。

  }
- mediaType：必选，MediaType类型，媒体类型，音频或视频。
- bitrate：必选，number类型，音视频流码率，单位为kbps。
- frameRate：必选，number类型，视频帧率，单位为帧/秒。

- rtt：必选，number类型，表示SDK到边缘服务器的RTT (Round-Trip Time)，单位毫秒。只有本地流才有rtt 值。
- jitter：必选，number类型，音视频流抖动值。
- pktLossRate：必选，number类型，音视频流丢包率。

}

8.5.4 流对象（Stream）

本章节介绍了Web SDK的Stream接口详情。

表 8-7 Stream 接口

接口	描述
play	播放该音视频流。
stop	停止播放视频流。
resume	恢复播放音视频。
close	关闭音视频。
muteAudio	禁用音频轨道。
muteVideo	禁用视频轨道。
unmuteAudio	启用音频轨道。
unmuteVideo	启用视频轨道。
getId	获取Stream唯一标识ID。
getUserId	获取Stream所属的用户ID。
setAudioOutput	设置音频输出设备。
setAudioVolume	设置音频音量大小。
getAudioLevel	获取实时音频音量级别。
hasAudio	流是否包含音频轨道。
hasVideo	流是否包含视频轨道。
getAudioTrack	获取流的音频轨道。
getVideoTrack	获取流的视频轨道。
getType	获取流类型。
on	注册流对象事件回调接口。
off	取消注册流对象事件回调接口。
getStreamInfo	获取流信息。
getVideoSnapshot	获取视频截图

play

```
async play(elementId: string, options?: Options): Promise<void>
```

【功能说明】

播放音视频流。该方法会自动创建<audio>、<video>标签，并在指定的标签上播放音频和视频，同时该标签会被添加到页面中名为 “elementId ”的div容器下。

【请求参数】

- elementId：必选，string类型，HTML 标签ID。
- options：可选，Options类型，播放选项。

  Options类型定义为：{
  - objectFit：可选，string类型，远端共享流的默认值为contain，其他流默认值为cover。支持的枚举值如下：
    - contain：优先保证视频内容全部显示。视频尺寸等比缩放，直至视频窗口的一边与视窗边框对齐。如果视频尺寸与显示视窗尺寸不一致，在保持长宽比的前提下，将视频进行缩放后填满视窗，缩放后的视频四周会有一圈黑边。
    - cover：优先保证视窗被填满。视频尺寸等比缩放，直至整个视窗被视频填满。如果视频长宽与显示窗口不同，则视频流会按照显示视窗的比例进行周边裁剪或图像拉伸后填满视窗。
    - fill：视频内容完全填充视窗。如果视频的宽高比与视窗不相匹配，那么视频将被拉伸以适应视窗。
  - muted：可选，boolean类型，true表示静音，false表示不静音。默认值为false。
  - resolutionId：可选，string类型。双流场景，指定播放的分辨率Id的视频，如果不指定，默认选择分辨率最高的视频。该参数为1.8.0版本新增。
  - }

【返回参数】

Promise<void>：返回一个Promise对象。

⚠ 注意

- 由于浏览器自动播放策略的限制，在play()返回错误后需要引导用户通过手动触发页面控件后，调用**resume**接口恢复播放。
- 本地流播放通常需要设置muted参数为true（静音），以防播放出来的声音也被麦克风采集到，造成回音的效果。
- 在App上，一个resolution对应于一个音视频播放窗口，Stream中的音频对所有的resolution是公共的。

stop

```
stop(option?: StopOption): void
```

【功能说明】

停止播放音视频流。

【请求参数】

option：可选，StopOption类型，表示停止播放的选项。如果不传，则停止该流的音频和所有分辨率的视频。

StopOption类型定义如下：{

- audio：可选，boolean类型。表示是否停止音频。默认为false。
- video：可选，boolean类型。表示是否停止视频。默认为false。
- resolutionIds：可选，string[] 型。在video为true的时候有效。指定停止播放的分辨率Id 的视频，如果不传resolutionIds，则默认停止所有分辨率的视频。

}

【返回参数】

无

resume

```
async resume(option?: ResumeOption): Promise<void>
```

【功能说明】

恢复播放音视频。场景说明如下：

- 在某些版本浏览器上，移动传入play()的div容器可能会导致音视频播放器进入PAUSED状态，此时需要调用该接口恢复播放。
- 由于浏览器自动播放策略的限制，在play()返回错误后需要引导用户通过手动方式调用该接口恢复播放。

【请求参数】

option：可选，表示恢复播放的选项，ResumeOption类型，如果不传则恢复播放该流的音频和所有分辨率的视频。

ResumeOption类型定义为：{

- audio：可选，boolean类型。是否resume音频。默认为false。
- video：可选，boolean类型。是否resume视频。默认为false。
- resolutionIds：可选，string[]类型，在video为true的时候有效。指定恢复播放的分辨率Id的视频，如果不传resolutionIds，默认恢复所有分辨率的视频。

}

【返回参数】

Promise<void>：返回一个Promise对象。

close

```
close(option?: CloseOption): void
```

【功能说明】

关闭音视频的播放。对于本地流，该方法在关闭播放的同时会关闭音视频采集，释放设备资源占用。

【请求参数】

option：可选，CloseOption类型，表示关闭音视频的选项。如果option不填写，则关闭音频和所有分辨率的视频。

CloseOption类型定义如下：{

- audio：可选，boolean类型。表示是否关闭音频。默认为false。
- video：可选，boolean类型。表示是否关闭视频。默认为false。
- resolutionIds：可选，string[]类型。在video为true的时候有效。指定关闭播放的分辨率Id的视频，如果不传resolutionIds，默认关闭所有分辨率的视频。

}

【 返回参数 】

无

muteAudio

```
muteAudio(): boolean
```

【 功能说明 】

禁用音频轨道。

【 请求参数 】

无

【 返回参数 】

boolean类型，true表示禁用音频轨道成功，false表示禁用音频轨道失败。

muteVideo

```
muteVideo(): boolean
```

【 功能说明 】

禁用视频轨道。

【 请求参数 】

无

【 返回参数 】

boolean类型，true表示禁用视频轨道成功，false表示禁用视频轨道失败。

unmuteAudio

```
unmuteAudio(): boolean
```

【 功能说明 】

启用音频轨道。

【 请求参数 】

无

【 返回参数 】

boolean类型，true表示启用音频轨道成功，false表示启用音频轨道失败。

unmuteVideo

```
unmuteVideo(): boolean
```

【 功能说明 】

启用视频轨道。

【 请求参数 】

无

【 返回参数 】

boolean类型，true表示启用视频轨道成功，false表示启用视频轨道失败。

getId

```
getId(): string
```

【 功能说明 】

获取Stream唯一标识ID。发布流中需包含视频流，发布后才能获取到有效的ID。

【 请求参数 】

无

【 返回参数 】

string类型，Stream唯一标识ID。

getUserId

```
getUserId(): string
```

【 功能说明 】

获取Stream所属的用户ID。对于本地流，如果在**createStream**的时候入参StreamConfig中没有设置该参数，则返回undefined。

【 请求参数 】

无

【 返回参数 】

string类型，Stream所属的用户ID。

setAudioOutput

```
setAudioOutput(deviceId: string): Promise<void>
```

【 功能说明 】

设置音频输出设备。

【 请求参数 】

deviceId：必选，string类型，音频输出设备的设备ID。

【 返回参数 】

Promise<void>：返回一个Promise对象。

setAudioVolume

```
setAudioVolume(volume: number): void
```

【 功能说明 】

设置音频音量大小。

【 请求参数 】

volume：必选，number类型，音量大小，取值范围为[0,100]。

【 返回参数 】

无

getAudioLevel

```
getAudioLevel(): number
```

【 功能说明 】

获取实时音量级别。

【 请求参数 】

无

【 返回参数 】

number类型，返回值在(0, 1)之间，通常该值大于0.1认为用户正在说话。

hasAudio

```
hasAudio(): boolean
```

【 功能说明 】

该Stream是否包含音频轨道。

【 请求参数 】

无

【 返回参数 】

boolean类型，true表示包含音频轨道，false表示不包含音频轨道。

hasVideo

```
hasVideo(): boolean
```

【 功能说明 】

该Stream 是否包含视频轨道

【 请求参数 】

无

【 返回参数 】

boolean类型，true表示包含视频轨道，false表示不包含视频轨道。

getAudioTrack

```
getAudioTrack(): MediaStreamTrack
```

【功能说明】

获取音频轨道

【请求参数】

无

【返回参数】

MediaStreamTrack类型。MediaStreamTrack类型详情可参见**MediaStreamTrack**。

getVideoTrack

```
getVideoTrack(resolutionId?:string): MediaStreamTrack
```

【功能说明】

获取视频轨道。

【请求参数】

resolutionId：可选，string类型。指定分辨率Id，如果不指定，默认选择分辨率最高的视频。

【返回参数】

MediaStreamTrack 类型。MediaStreamTrack类型详情可参见**MediaStreamTrack**。

getType

```
getType(): string
```

【功能说明】

获取流类型。用于判断一个流是主流还是辅流，辅流通常是一个屏幕共享流。

【请求参数】

无

【返回参数】

string类型，本地流：local，远端主流：main，远端辅流：auxiliary。

on

```
on(event: string, handler: function): void
```

【功能说明】

注册流对象事件回调接口。

【请求参数】

- event：必选，string类型，事件名称。详细事件列表请参见**RTCStreamEvent**。
- handler：必选，function类型，事件处理方法。

【返回参数】

无

off

```
off(event: string, handler: function): void
```

【功能说明】

取消注册流对象事件回调接口。

【请求参数】

- event：必选，string类型，事件名称。详细事件列表请参见**RTCStreamEvent**。
- handler：必选，function类型，事件处理方法。

【返回参数】

无

getStreamInfo

```
getStreamInfo(): StreamInfo
```

【功能说明】

获取已经初始化的本地流，或者收到的远端流的的信息。

【请求参数】

无

【返回参数】

StreamInfo 类型定义如下：{

- videoProfiles：RTCVideoProfileInfo[]类型。
- audioProfile：RTCAudioProfile类型。

}

RTCVideoProfileInfo类型定义如下：{

- resolutionId：string类型，表示分辨率Id。
- hasTrack：boolean类型，表示该分辨率的视频是否具备可播放的track。
- width：number类型，分辨率的宽度，单位为像素。
- height：number类型，分辨率的高度，单位为像素。
- frameRate：number类型，视频帧率，单位为帧/秒。
- minBitrate：number类型，视频最小码率，单位为bps。
- maxBitrate：number类型，视频最大码率，单位为bps。

}

RTCAudioProfile类型定义如下：{

- sampleRate：number类型，表示音频采样率。
- channelCount：number类型，表示音频声道数。
- bitrate：number类型，表示音频码率，单位为bps。

```
}
```

getVideoSnapshot

```
getVideoSnapshot(resolutionId?: string,format?: string, encoderOptions?: number): string
```

【功能说明】

获取视频截图

【请求参数】

- resolutionId：可选，string类型。双流场景，指定需要截图的视频分辨率Id，resolutionId可通过接口**getStreamInfo** 获取。如果不传resolutionId，则默认对分辨率最高的视频流截图。
- format：可选，string类型。表示截图的格式，可填‘jpeg’、‘webp’、'png'。如果不传format，则默认使用png格式截图。
- encoderOptions：可选，number类型。format填‘jpeg’或者‘webp’时生效，表示图片质量，值范围【0,1】，默认值为1。

【返回参数】

string类型，表示视频截图对象的url，当视频未播放，或者resolutionId错误时，返回的string为空。

8.5.5 本地流对象（LocalStream）

该对象继承自Stream对象，并有如下新增接口。

表 8-8 LocalStream 接口

接口	描述
initialize	本地流初始化。
setAudioProfile	设置音频流配置。
setVideoProfile	设置视频流配置。
setScreenProfile	设置辅流配置。
addAudioTrackCapture	流初始化后，流对象中没有音频track的，可通过该接口增加音频track采集。该接口为1.10.0版本新增。
addVideoTrackCapture	流初始化后，流对象中没有视频track的，通过该接口可增加视频track采集。该接口为1.10.0版本新增。
addResolution	对初始化后的本地流增加新的分辨率的视频。
removeResolution	对流删除视频指定分辨率的视频。
addTrack	为本地流对象添加音视频轨。
removeTrack	从本地流对象移除音视频轨。
replaceTrack	为本地流对象替换音视频轨。
switchDevice	切换媒体输入设备。

接口	描述
startAudioMixing	开始播放在线音频文件。
stopAudioMixing	停止播放在线音频文件。
pauseAudioMixing	暂停播放在线音频文件。
resumeAudioMixing	恢复播放在线音频文件。
getAudioMixingDuration	获取在线音频文件时长。
setAudioMixingVolume	设置在线音频音量大小。
setAudioMixingPosition	设置在线音频播放进度。
getAudioMixingCurrentPosition	获取在线音频播放进度。
bindScreenAudio2RelatedStream	绑定屏幕共享背景音至关联流对象。该接口为1.4.0版本新增。

initialize

```
initialize(): Promise<StreamInitializeResult>
```

【功能说明】

根据createStream入参StreamConfig，初始化本地音视频流对象。流只有初始化完成后才可以播放。

【请求参数】

无

【返回参数】

流初始化结果：StreamInitializeResult类型。

⚠ 注意

StreamInitializeResult表示流初始化的结果。对象继承自RtcError。提供如下方法：

- 获取所有媒体类型的初始化结果：
  getMediaCaptureResult(): MediaCaptureResult[]
- 获取指定媒体类型的初始化结果：
  getMediaCaptureResultByType(type: MediaType): MediaCaptureResult

MediaCaptureResult类型定义如下：

{

- type：必选，表示媒体类型，MediaType类型，MediaType枚举值包括audio或者video。
- track：可选，表示对应媒体类型如果初始化成功后的 track，MediaStreamTrack类型。
- error：可选，表示对应媒体类型如果初始化失败的错误，RtcError类型。

}

如果初始化音视频采集失败，则结果中会返回对应的错误信息。错误码范围90000001，90100017~90100020，具体请参考表8-13。

setAudioProfile

setAudioProfile(profile: string|RTCAudioProfile): void

【功能说明】

设置音频流采样率、声道和码率等配置。如果未调用该接口设置，则SDK设置为默认值standard。

【请求参数】

profile：必选，string类型或RTCAudioProfile类型。 若为string类型，则相关的采样率、声道数、码率如表8-9所示。 若为RTCAudioProfile类型，则需要根据实际情况进行配置，推荐使用已定义的profile，输入不合法时，默认使用standard。

表 8-9 profile 对应的采样率、声道数和码率

profile	采样率(千赫兹/KHZ)	声道数	码率（Kbps）
low	16	1	24
standard	48	1	40
high	48	1	128

RTCAudioProfile类型定义为：{

- sampleRate：可选，number类型，采样率。
- channelCount：可选，number类型，音轨数。
- bitrate：可选，number类型，码率，单位为bps。

}

【返回参数】

无

注意

该方法需要在调用**initialize**之前调用。

需注意码表中单位为kbps，接口传参的单位为bps，实际调用接口时需进行转换。

setVideoProfile

```
setVideoProfile(profile: string|RTCVideoProfile,resolutionId?: string): void
```

【功能说明】

设置视频主流配置，如分辨率、帧率和码率等。如果未调用该接口设置，则SDK设置默认值为360p_2。如果该流已经发布，则该流会自动重新发布到远端。

【请求参数】

- profile：必选，string类型或RTCVideoProfile类型。 若为string类型，则相关的分辨率、帧率和码率如**表8-10**。 若为RTCVideoProfile类型，则需要根据实际情况进行配置，推荐使用已定义的“profile”。输入不合法时，默认使用360p_2。

表 8-10 profile 对应的分辨率、帧率和码率

profile	分辨率	帧率	最小码率（kbps）	最大码率（kbps）
90p_1	160 x 90	15	64	110
90p_2	120 x 90	15	64	110
120p_1	160 x 120	15	64	120
120p_2	120 x 120	15	64	110
180p_1	320 x 180	15	80	320
180p_2	240 x 180	15	80	170
180p_3	180 x 180	15	64	130
240p_1	320 x 240	15	100	400
240p_2	240 x 240	15	80	320
270p_1	480 x 270	15	160	600
300p_1	400 x 300	15	200	500
360p_1	640 x 360	15	200	800
360p_2	480 x 360	15	200	700

profile	分辨率	帧率	最小码率（kbps）	最大码率（kbps）
360p_3	360 x 360	15	150	600
450p_1	800 x 450	15	300	950
480p_1	640 x 480	15	250	900
480p_2	480 x 480	15	200	800
540p_1	960 x 540	15	400	1000
630p_1	1120 x 630	15	450	1150
720p_1	1280 x 720	15	500	1500
720p_2	960 x 720	15	450	1100
1080p_1	1920 x 1080	15	600	2000
1080p_2	1440 x 1080	15	550	1700

- resolutionId：可选，string类型。在双流场景下，指定要设置的分辨率Id的视频，如果不指定，默认选择最高的分辨率的视频。

RTCVideoProfile类型定义为：{

- width：可选，number类型，分辨率的宽度，单位为像素。
- height：可选，number类型，分辨率的高度，单位为像素。
- frameRate：可选，number类型，视频帧率，单位为帧/秒。
- minBitrate：可选，number类型，视频最小码率，单位为bps。
- maxBitrate：可选，number类型，视频最大码率，单位为bps。

}

【返回参数】

无

⚠ 注意

- 需注意码表中单位为kbps，接口传参的单位为bps，实际调用接口时需进行转换。
- 自定义采集的流（使用videoSource创建的Stream）不支持动态调用此接口，仅支持摄像头采集的流调用。
- 由于设备采集能力、系统性能以及浏览器的限制，视频分辨率，帧率，码率的实际值不一定能够完全匹配设定值，这种情况下浏览器会自动调整分辨率，尽可能匹配设定值，具体分辨率以实际采集到的分辨率为准。
- 能否采集1080p及以上的分辨率取决于采集设备能力以及系统性能，iOS Safari不支持采集1080p。
- 码表中的1080p分辨率为2.0.0版本新增，1.0版本如需采集1080P分辨率，请使用自定义参数的方式设置。
- Firefox不支持自定义帧率（默认为30fps）。
- 如果上表的Profile不符合您的要求，用户可以根据自身业务需求自定义Profile。

setScreenProfile

```
setScreenProfile(profile: string|RTCScreenProfile): void
```

【功能说明】

设置辅流配置，包括分辨率、帧率和码率等。如果未调用该接口设置，则SDK设置默认值为720p。

【请求参数】

profile：必选，string类型或RTCScreenProfile类型。若为string类型，则相关分辨率、帧率、码率如表8-11所示。如果为RTCScreenProfile，则需要根据实际情况进行配置，推荐使用已定义的“profile”。输入不合法时，默认使用720p。

表 8-11 profile 对应的分辨率、帧率和码率

profile	分辨率	帧率	码率（kbps）
720p	1280 x 720	15	1200
1080p	1920 x 1080	15	2000

RTCScreenProfile类型为: {

- width：可选，number类型，分辨率的宽度，单位为像素。
- height：可选，number类型，分辨率的高度，单位为像素。
- frameRate：可选，number类型，视频帧率，单位为帧/秒。
- bitrate：可选，number类型，视频码率，单位为bps。

}

【返回参数】

无

⚠ 注意

该方法需要在调用**initialize**之前调用。

需注意码表中单位为kbps，接口传参的单位为bps，实际调用接口时需进行转换。

addAudioTrackCapture

```
addAudioTrackCapture(microphoneId?: string): promise<MediaStreamTrack>
```

【功能说明】

已经初始化的本地流对象，如果未初始化音频或者音频初始化失败，可以调用该接口对本地流增加音频的track采集。若该本地流已经被发布，则该流会自动重新发布到远端。该接口为1.10.0版本新增。

【请求参数】

microphoneId：可选，string类型，指定采集的麦克风设备Id。如果不传，SDK采用**createStream**入参StreamConfig中指定的microphoneId。

【返回参数】

MediaStreamTrack类型。表示增加成功的track。

⚠ 注意

在通过**removeTrack**移除音频的track后，可以通过该接口重新添加音频的track。

addVideoTrackCapture

```
addVideoTrackCapture(option?: VideoCaptureOption): promise<MediaStreamTrack>
```

【功能说明】

已经初始化的本地流对象，如果未初始化视频或者视频初始化失败，可以调用该接口对本地流增加视频的track采集。若该本地流已经被发布，则该流会自动重新发布到远端。该接口为1.10.0版本新增。

【请求参数】

option：可选，指定采集参数。VideoCaptureOption类型。

VideoCaptureOption定义如下：{

- cameraId：可选，string类型。指定摄像头设备Id。对于Android设备：user表示前置摄像头，environment表示后置摄像头。如果不传，SDK采用**createStream**入参StreamConfig中指定的cameraId和facingMode 。
- resolutionId：可选，string类型。指定要设置的分辨率Id的视频。如果不传，SDK默认选择分辨率最高的视频。

}

【返回参数】

MediaStreamTrack类型。表示增加成功的track。

⚠ 注意

在通过**removeTrack**移除视频的track后，可以通过该接口重新添加视频的track。

addResolution

```
addResolution(profile: string|RTCVideoProfile,audio?: boolean): promise<RTCVideoProfileInfo>
```

【功能说明】

为已经初始化的本地流对象添加指定分辨率的视频。若该本地流已经被发布，则该流会自动重新发布到远端。

【请求参数】

- profile：必选，string类型，表示添加的分辨率视频的参数信息。RTCVideoProfile类型说明参考**setVideoProfile**部分的说明。
- audio：可选，boolean类型，表示是否创建音频，true表示创建，false表示不创建。Stream中的音频对该stream中的所有的resolution是公共的。如果创建流的配置开启了音频但是没有音频track，则默认创建。

【返回参数】

RTCVideoProfileInfo类型。增加成功后resolution的profile信息。RTCVideoProfileInfo定义参考**getStreamInfo**接口定义。

如果失败，则返回StreamInitializeResult。StreamInitializeResult参考**initialize**接口的定义。

⚠ 注意

- 当前对一个LocalStream最多可支持2个分辨率。
- 当开启大小流的场景下，流的分辨率设置接口**setVideoProfile**使用会存在限制（两条流的分辨率大者，决定了可调整的最大分辨率上限），建议谨慎使用。

removeResolution

```
removeResolution(resolutionId: string): promise<void>
```

【功能说明】

对本地流对象删除视频分辨率。若该本地流已经被发布，则该流会自动重新发布到远端。

【请求参数】

resolutionId：必选。string类型，指定要删除的视频的分辨率Id。

【返回参数】

Promise<void>：返回一个Promise对象。

addTrack

```
addTrack(track: MediaStreamTrack,resolutionId?: string ): promise<void>
```

【功能说明】

为初始化后的本地流对象添加音视频轨。若该本地流已经被发布，则该流会自动重新发布到远端。

【请求参数】

- track：必选，MediaStreamTrack类型。指定要添加的track。
- resolutionId：可选，string类型。在双流主流场景下，指定分辨率Id的视频，如果不指定，默认选择分辨率最高的视频增加 track。

【返回参数】

Promise<void>：返回一个Promise对象。

⚠ 注意

- 如果分辨率的视频track已经存在，则不能重复添加。
- 如果需要更新旁路推流等操作，需要等待异步操作完成

removeTrack

```
removeTrack(track: MediaStreamTrack): promise<void>
```

【功能说明】

从本地流对象移除音视频轨。若该本地流已经被发布，则该流会自动重新发布到远端。

【请求参数】

track：必选，MediaStreamTrack类型。指定要移除的track。

【返回参数】

Promise<void>：返回一个Promise对象。

⚠ 注意

- 如果麦克风对应的音频track都移除，则SDK不会再访问该麦克风。
- 如果摄像头对应的视频track都移除，则SDK不会再访问该摄像头。
- 如果需要更新旁路推流等操作，需要等待异步操作完成

replaceTrack

```
replaceTrack(track: MediaStreamTrack, type: "audio" | "video",resolutionId?: string): promise<void>
```

【功能说明】

为初始化后的本地流对象替换音视频轨。若该本地流已经被发布，则该流会自动重新发布到远端。

【请求参数】

- track：必选，MediaStreamTrack类型。
- type：必选，string类型，"audio" | "video"。
- resolutionId：可选，string类型。type为video的时候有效。指定要替换的分辨率Id的视频，如果不指定，默认选择分辨率最高的视频。

【 返回参数 】

Promise<void>：返回一个Promise对象。

switchDevice

```
switchDevice(deviceType: "audio" | "video", deviceId: string): Promise<void>
```

【 功能说明 】

切换媒体输入设备。若该本地流已经被发布，则会自动更新发布到远端的音视频流。

【 请求参数 】

- deviceType：必选，string类型，"audio" | "video"。
- deviceId：必选，string类型， 输入设备的设备ID。

【 返回参数 】

Promise<void>：返回一个Promise对象。

startAudioMixing

```
startAudioMixing(option: AudioMixOption): Promise<void>
```

【 功能说明 】

开始播放在线音频文件。

【 请求参数 】

option：必选，音频文件播放参数。参数类型为AudioMixOption。

AudioMixOption定义为：{

- filePath：必选，string类型，表示在线音频文件的下载路径。支持音频格式包括MP3、AAC以及浏览器支持的其他音频格式。
- startTime：可选，number类型，表示音频文件开始播放的时间点，默认值为0。
- replace：可选，boolean类型，如果为true表示要用音频文件替换本地音频流，默认值为false。
- loop：可选，boolean类型，如果为true表示需要无限循环播放，默认值为false。
- repeatCount：可选，number类型，循环播放次数。如果为0，表示不重复播放，默认值为0。

}

【 返回参数 】

Promise<void>：返回一个Promise对象。

⚠ 注意

- loop设置为true时，repeatCount需要设置为0。
- 混音功能相关接口仅在**publish**接口调用成功后方可调用。

stopAudioMixing

```
stopAudioMixing(): Promise<void>
```

【功能说明】

停止播放在线音频文件。

【请求参数】

无

【返回参数】

Promise<void>：返回一个Promise对象。

pauseAudioMixing

```
pauseAudioMixing(): void
```

【功能说明】

暂停播放在线音频文件。

【请求参数】

无

【返回参数】

无

resumeAudioMixing

```
resumeAudioMixing(): void
```

【功能说明】

恢复播放在线音频文件。

【请求参数】

无

【返回参数】

无

getAudioMixingDuration

```
getAudioMixingDuration(): number
```

【功能说明】

获取在线音频文件时长。单位为毫秒。

【请求参数】

无

【 返回参数 】

number类型，音频文件时长。

setAudioMixingVolume

```
setAudioMixingVolume(level: number): void
```

【 功能说明 】

设置在线音频音量大小。

【 请求参数 】

level：必选，number类型，表示音量大小。

【 返回参数 】

无

setAudioMixingPosition

```
setAudioMixingPosition(position: number): void
```

【 功能说明 】

设置在线音频播放进度。

【 请求参数 】

position：必选，number类型，表示进度，取值不能大于音频实际时长。单位为毫秒。

【 返回参数 】

无

getAudioMixingCurrentPosition

```
getAudioMixingCurrentPosition(): number
```

【 功能说明 】

获取在线音频播放进度。

【 请求参数 】

无

【 返回参数 】

number类型，当前音频播放进度。单位为毫秒。

bindScreenAudio2RelatedStream

```
bindScreenAudio2RelatedStream(bindStream: LocalStream, muteMainStreamAudio?: boolean): void
```

【 功能说明 】

绑定屏幕共享背景音至关联流对象。

【 请求参数 】

- bindStream：必选，LocalStream类型, 创建并初始化成功的本地主流对象。
- muteMainStreamAudio：可选，boolean类型, 主流音频是否静音，默认值为false。true表示主流音频静音，false表示主流音频不静音。

【返回参数】

无

⚠ 注意

- 该接口仅由通过**HRTC.createStream**创建的辅流对象调用。
- 屏幕共享背景音需要借助主流的音频通道发送，若想要订阅共享背景音，至少需要订阅主流音频。
- 弹出的共享选择窗口，需要勾选左下角的共享音频复选框，否则屏幕共享背景音将无法共享。
- 该功能仅支持Windows平台Chrome浏览器74及以上版本。

8.5.6 远端流对象（RemoteStream）

该对象继承自Stream对象。

8.5.7 流事件通知（RTCStreamEvent）

本章节介绍了Web SDK的RTCStreamEvent事件。

表 8-12 StreamEvent 事件

接口	描述
player-state-change	播放状态变更事件。
screen-sharing-stopped	共享屏幕停止事件。
audio-mixing-played	本地混音播放事件。
audio-mixing-finished	本地混音播放结束事件。

⚠ 注意

事件注册监听应在业务结束时取消注册，否则注册监听事件累积会有内存泄漏风险。

player-state-change

【事件说明】

播放状态变更事件。在播放状态变更的时候触发。

【回调参数】

event：playState类型。字段定义如下：

- type：string类型，表示播放器类型，取值为video/audio。
- id：string类型，表示流分辨率Id。
- state：string类型，表示当前播放状态。取值包括：PLAYING，STOPPED，PAUSED， NONE。
- reason：string类型，表示触发播放状态变更的原因。

screen-sharing-stopped

【事件说明】

共享屏幕停止事件。仅在本地共享屏幕停止时触发。

【回调参数】

event：string类型。表示停止共享屏幕时的流Id。

audio-mixing-played

【事件说明】

混音播放事件。仅在本地混音播放时触发。

【回调参数】

无

audio-mixing-finished

【事件说明】

混音播放结束事件。仅在本地混音结束播放时触发。

注意：手动调用**stopAudioMixing**和**pauseAudioMixing**不会触发此事件。

【回调参数】

无

8.5.8 错误码（RtcError）

getCode

```
getCode(): number
```

【功能说明】

获取错误码。

【请求参数】

无

【返回参数】

number类型，错误码值。

getMsg

```
getMsg(): string
```

【 功能说明 】

获取错误描述。

【 请求参数 】

无

【 返回参数 】

string类型，错误码描述。

8.5.9 客户端错误码

本章节介绍了Web SDK的客户端错误码RtcErrorCode的详细信息。

表 8-13 错误码说明

类成员	错误码	描述	错误原因或建议处理方式
RTC_ERR_CODE_SUCCESS	0	success	成功。
RTC_ERR_CODE_RTC_SDK_ERROR	90000001	sdk internal error	SDK内部错误，请联系技术支持。
RTC_ERR_CODE_WAIT_RSP_TIMEOUT	90000004	message response timeout	消息响应超时，请联系技术支持。
RTC_ERR_CODE_INVALID_PARAMETER	90000005	invalid parameter	参数传递错误，请参照API文档排查。
RTC_ERR_CODE_INVALID_OPERATION	90100001	illegal operation	非法操作，用户状态不正确。
RTC_ERR_CODE_NOT_SUPPORT_MEDIA_DEVICES	90100002	not support enumerate devices	浏览器不支持enumerateDevices方法。
RTC_ERR_CODE_NO_AVAILABLE_DEVICES	90100003	no available devices	没有找到可用设备，请排查设备是否就绪。
RTC_ERR_CODE_NO_AVAILABLE_VIDEO_INPUT_DEVICES	90100004	no available video input devices	没有找到可用摄像头设备，请排查视频采集设备是否就绪。
RTC_ERR_CODE_NO_AVAILABLE_AUDIO_INPUT_DEVICES	90100005	no available audio input devices	没有找到音频输入设备，请排查音频采集设备是否就绪。
RTC_ERR_CODE_NO_AVAILABLE_AUDIO_OUTPUT_DEVICES	90100006	no available audio output devices	没有找到音频输出设备。
RTC_ERR_CODE_STATUS_ERROR	90100007	room status error	房间状态不正确，检查是否入会成功。

类成员	错误码	描述	错误原因或建议处理方式
RTC_ERR_CODE_WEBSOCKET_NOT_CONNECTED	90100008	websocket connection state is not "CONNECTED"	websocket 链接未成功，检查链接情况。
RTC_ERR_CODE_WAIT_CONFIG_FAIL	90100009	wait server config fail	获取下发配置失败，请联系技术支持。
RTC_ERR_CODE_PUBLISH_RESPONSE_FAIL	90100010	publish response fail	发布流响应失败，请联系技术支持。
RTC_ERR_CODE_REGION_NOT_COVERED	90100011	current region is not covered, service unavailable	没有找到服务端地址，请联系技术支持。
RTC_ERR_CODE_WEBSOCKET_CONNECT_TIMEOUT	90100012	websocket connect timeout	websocket建链超时，请联系技术支持。
RTC_ERR_CODE_WEBSOCKET_RECONNECT_TIMEOUT	90100013	websocket reconnect timeout	websocket重连超时，请联系技术支持。
RTC_ERR_CODE_WEBSOCKET_NOT_OPEN	90100014	websocket is not open	websocket链接未打开，请联系技术支持。
RTC_ERR_CODE_WEBSOCKET_INTERRUPTED	90100015	websocket connection state is idle, interrupt operation	websocket链接被强制关闭，一般为离会或者重连。
RTC_ERR_CODE_WEBSOCKET_CONNECT_ERROR	90100016	websocket connect error	websocket监听onerror，服务端主动断链。
RTC_ERR_CODE_CAPTURE_PERMISSION_DENIED	90100017	capture failed, permission denied	采集失败，音视频设备采集权限未被授权。建议提示用户授权摄像头/麦克风访问权限。
RTC_ERR_CODE_CAPTURE_OVER_CONSTRAINED	90100018	capture failed, Constraint parameter invalid	采集失败，音视频采集设备不支持设置的采集约束。
RTC_ERR_CODE_CAPTURE_DEVICE_NOT_FOUND	90100019	capture failed, requested device not found	采集失败，设备未找到。建议在通话开始前引导用户检查通话所需的摄像头或麦克风等设备是够就绪。

类成员	错误码	描述	错误原因或建议处理方式
RTC_ERR_CODE_CAPTURE_DEVICE_NOT_READABLE	90100020	capture failed, maybe device is occupied by other application	采集失败，设备被占用，请检查使用状态。建议提示用户“暂时无法访问摄像头/麦克风，请确保当前没有其他应用请求访问摄像头/麦克风，并重试”。
RTC_ERR_CODE_PLAY_NOT_ALLOW	90100021	the user didn't interact with the document first, please trigger by gesture	不允许播放。
RTC_ERR_CODE_ROLE_NO_PERMISSION	90100022	the user role have no permission to operate	用户角色没有权限，请检查用户角色。
RTC_ERR_CODE_ANSWER_SDP_INVALID	90100023	the answer sdp is invalid	SDP协商错误，请联系技术支持。
RTC_ERR_CODE_MEDIA_UPSTREAM_UNSUPPORTED	90100024	the upstream media is not supported	浏览器不支持媒体采集。
RTC_ERR_CODE_MEDIA_NETWORK_ERROR	90100026	media connection establish failed, please switch network or try again later	媒体建链失败，请切换网络重试。
RTC_ERR_CODE_CLIENT_RELAY_ROOM_OVER_MAXNUM	90100027	relay room number over maxium number	跨房数量超过最大值。
RTC_ERR_CODE_CLIENT_RELAY_JOINER_OVER_MAXNUM	90100028	joiner already exist in relay rooms	跨房内主播人数超过最大值。
RTC_ERR_CODE_ROOM_STREAM_STATUS_PAUSED	90100029	room stream status paused	房间音视频暂停。
RTC_ERR_CODE_SIGNATURE_EXPIRED	90100030	signature expired	签名过期。
RTC_ERR_CODE_SIGNATURE_INVALID	90100031	signature invalid	签名非法。

类成员	错误码	描述	错误原因或建议处理方式
RTC_ERR_CODE_RTC_ACS	90100100	server internal exception	服务端内部错误，请联系技术支持。
RTC_ERR_CODE_RTC_CONTROL_ERROR	90100200	server internal exception	服务端内部错误，请联系技术支持。
RTC_ERR_CODE_SFU_ERROR	90100600	server internal exception	服务端内部错误，请联系技术支持。

8.5.10 服务端错误码

当SDK运行出现网络、媒体相关等错误时，SDK无法自动恢复，需要APP干预或进行用户提示。该错误码由服务端产生，通过onError返回。

表 8-14 服务端错误码

错误码	描述	错误原因
RTC.10000001	内部错误	程序或环境问题
RTC.31000000	节点不存在	程序或环境问题
RTC.31000001	session校验失败	程序或环境问题
RTC.31000003	内部异常	程序或环境问题
RTC.31000004	认证失败	用户使用问题
RTC.31000005	请重试	用户使用问题
RTC.31000006	需要时钟同步	用户使用问题
RTC.31000007	请求资源不存在	程序或环境问题
RTC.32000000	服务异常	程序或环境问题
RTC.32000001	流号已满	程序或环境问题
RTC.32000002	SFU为空	程序或环境问题
RTC.32000004	下发流信息失败	程序或环境问题
RTC.32000005	添加适配器失败	程序或环境问题
RTC.32000006	添加路由失败	程序或环境问题
RTC.32000007	获取用户信息异常	程序或环境问题
RTC.32000010	选看用户不存在	程序或环境问题
RTC.32000011	音频速率参数非法	程序或环境问题
RTC.32000012	用户列表为空	程序或环境问题
RTC.32000013	非法请求参数	程序或环境问题

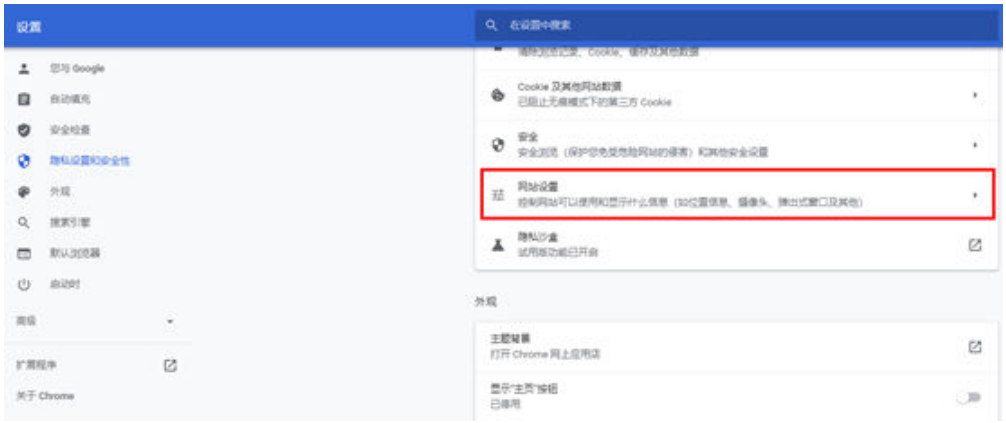
错误码	描述	错误原因
RTC.32000015	内部调用异常	程序或环境问题
RTC.32000016	内部调用异常	程序或环境问题
RTC.32000017	站点不存在	程序或环境问题
RTC.32000018	错误的加密算法	程序或环境问题
RTC.32000019	客户端媒体加密密钥base64解码失败	程序或环境问题
RTC.32000020	生成媒体加密密钥失败	程序或环境问题
RTC.32000021	下发加密信息异常	程序或环境问题
RTC.32000022	获取SFU的IP失败	程序或环境问题
RTC.32000024	内部调用异常	程序或环境问题
RTC.32000025	内部调用异常	程序或环境问题
RTC.32000028	不支持的操作	程序或环境问题
RTC.32000030	sfu资源不足	程序或环境问题
RTC.32000032	跨房数量超过上限	用户使用问题
RTC.32000033	不允许重复跨入同一房间	用户使用问题
RTC.32000034	稍后重试	程序或环境问题
RTC.33000000	服务异常	程序或环境问题
RTC.33000001	节点不存在	程序或环境问题
RTC.34000001	房间已满	用户使用问题
RTC.34000002	房间不存在	程序或环境问题
RTC.34000003	站点不存在	程序或环境问题
RTC.34000004	内部调用异常	程序或环境问题
RTC.34000006	用户不存在	用户使用问题
RTC.34000007	已存在辅流共享	用户使用问题

8.5.11 授权浏览器摄像头/麦克风访问权限的方法

谷歌浏览器

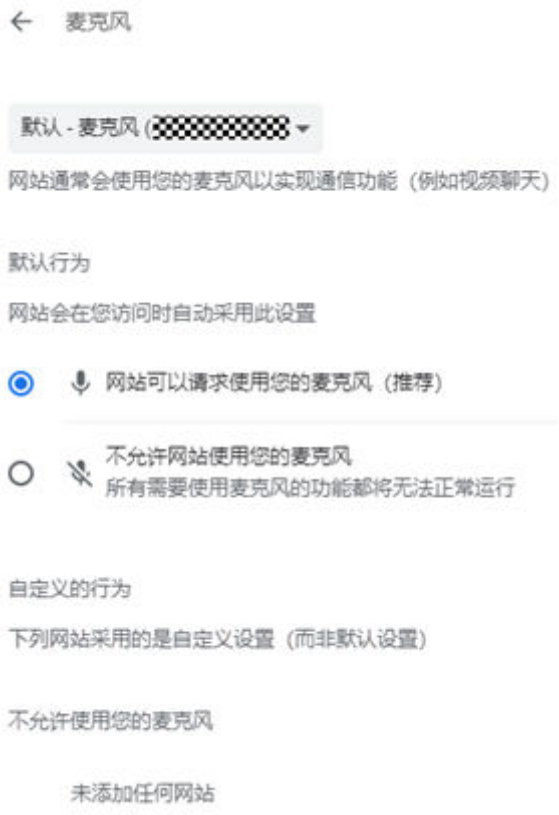
1. 打开谷歌浏览器，单击右上角设置图标。

2. 单击“设置”，打开设置页面。选择“隐私设置和安全性”，再单击“网站设置”。



3. 进入网站设置页面，分别单击“摄像头”和“麦克风”。

4. 进入摄像头、麦克风授权页面，选择网站可以请求使用您的摄像头、麦克风权限即可。

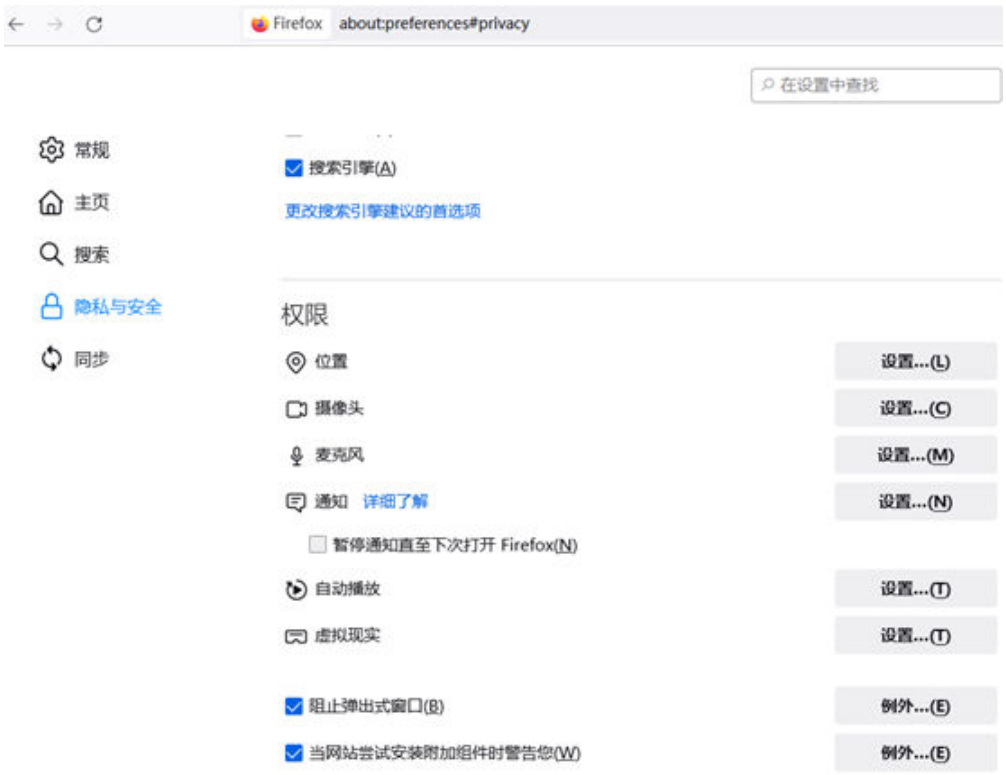


5. 完成以上设置后，浏览器在需要使用摄像头、麦克风时，就会在页面弹出设备询问框，选择“允许”即可。

火狐浏览器

1. 打开火狐浏览器，单击右上角设置图标，单击“设置”。

2. 打开设置页面，单击“隐私与安全”，找到摄像头、麦克风权限。分别单击“摄像头”和“麦克风”的“设置”。



3. 进入设置页面，将请求使用摄像头、麦克风权限的网站加入使用列表，并单击“保存更改”。

4. 完成以上设置后，浏览器在需要使用摄像头、麦克风时，就会在页面弹出设备询问框，选择“允许”即可。

搜狗浏览器

1. 打开搜狗浏览器，单击右上角设置图标，单击“选项”。

2. 打开选项页面，单击“高级”，找到“隐私保护”，再单击“内容设置”。

3. 进入内容设置页面，选择当前网站使用您的摄像头、麦克风时询问您，单击“完成”即可。

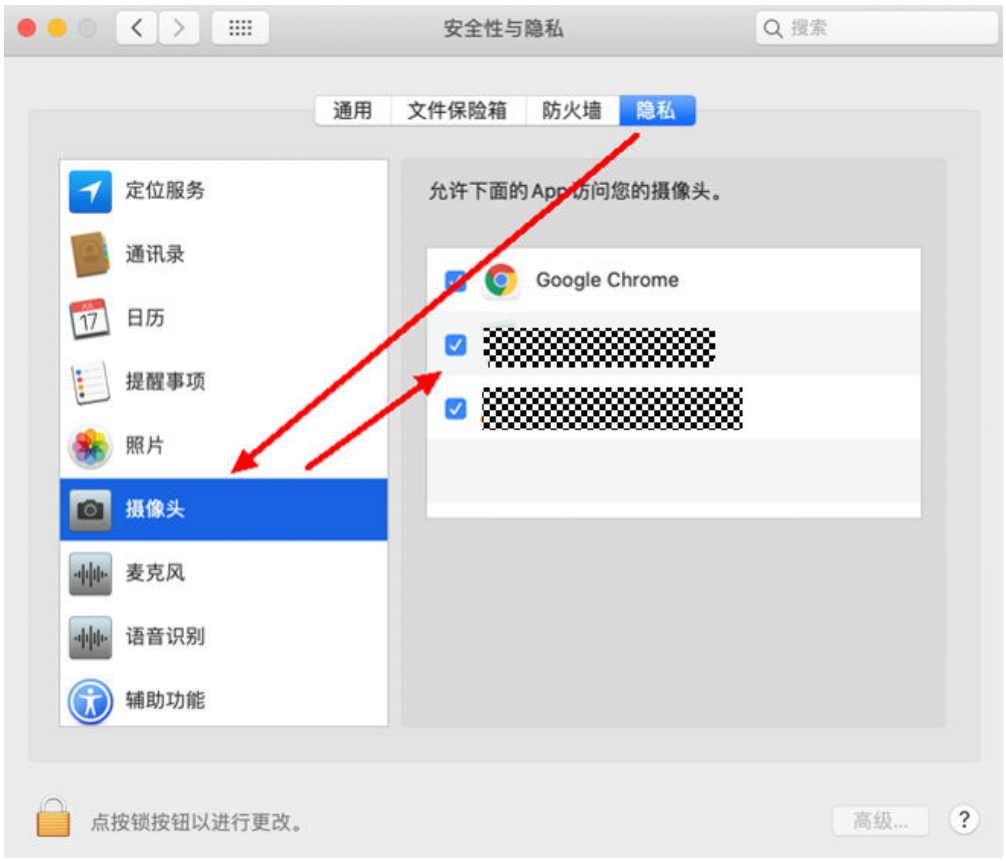
4. 完成以上设置后，浏览器在需要使用摄像头、麦克风时，就会在页面弹出设备询问框，选择“允许”即可。

Mac 系统的浏览器

1. 在程序坞中找到“系统偏好设置”并单击图标。

2. 打开后找到“安全性与隐私”并单击打开。

3. 在“安全性与隐私”页面选择“隐私”，然后单击“摄像头”/“麦克风”，将需要使用摄像头/麦克风的应用设置为允许。



8.6 常见问题

- **加入房间时，userName必须填吗?**

  非必填。userName、userId由App自定义，但可以相同。

  - userId：必填，string[64]类型，用户标识，userId需要保证应用内唯一。userId支持的字符包括：a-z、A-Z、0-9、连接符 '-'、下划线 '_'。
  - userName：选填，string[128]类型，用户昵称。

- **microphoneId跟cameraId在哪获取? 为什么是必填的?**

  分别为麦克风ID和摄像头ID ，音视频通话必须的，即在创建流的时候采集ID对应的音频、视频。

  使用**getDevices**、**getCameras**、**getMicrophones**接口可直接获取媒体输入输出、麦克风/摄像头设备ID 。

- **若摄像头不打开，是否显示默认的人像图标?**

  如果获取不到摄像头 ，但是能获取到麦克风，则视频是黑屏，音频有流。如果麦克风和摄像头都获取不到，则本地预览失败，不会显示默认的人像图标。

- **退出房间后摄像头没关，是不是需要释放摄像头?**

  退出房间不需要手动释放摄像头，会自动关闭，不再采集摄像头。

- **如何鉴权? 在什么时候鉴权?**

  具体请参见**接入鉴权**。

- **客户端以joiner角色加入房间失败。**

  客户端调用**join**时传入的角色参数不对。role是数值型，如果客户端传入的是字符串型，则会加入房间失败。

- **创建本地流失败，控制台提示Cannot read property 'getUserMedia' of undefined，无法获取到媒体源。**

  可能有以下原因：

  原因1：系统未允许应用访问摄像头等媒体源，解决方法请参见**授权浏览器摄像头/麦克风访问方法**。

  原因2：由于浏览器的策略，仅允许通过https://方式或者localhost的方式访问用户的摄像头和麦克风权限。

  原因3：检查摄像头等设备是否被其他应用占用了。

- **如果之前访问过使用Web SDK开发的App网站，又清理了该网站的权限，存在一定几率无法开启摄像头和麦克风。**

  在保证打开**授权浏览器摄像头/麦克风访问权限的方法**前提下。点开网页的左上角，将权限改为允许。如下图所示：

- **输入在线音频地址，且该地址可以在浏览器中打开，但是使用Web SDK的混音功能时，启动混音失败。**

  需要确认在线音频文件下载服务器是否支持跨域，由于浏览器的安全策略，必须要支持跨域，否则请求失败。

  注意：混音只有对端可以听到，本端听不见。

- **音频TopN 模式（音频最大三方模式）是什么意思?**

  音频TopN模式也叫音频最大三方模式。开启音频TopN模式后，本地用户不需要通过调用接口，单独订阅某个远端用户的音频，即可接收到当前房间内音量值最大的三个用户的音频。具体接口调用可参见**切换音频模式**。

- **调用setVolume4TopThree设置音频最大三方音量值是设置房间内三个最大声音的用户的音量值吗？传参需要传一个参数还是三个参数?**

  是的。传参只需要传一个参数。

- **如果业务上App只能使用http协议，是否能够集成使用SparkRTC Web SDK？**

  可以集成使用，但不推荐。需要用户手动关闭安全策略相关的开关。打开chrome页签，输入**chrome://flags/#unsafely-treat-insecure-origin-as-secure**，开启开关项，并把App的加载地址加入到忽略列表。

- **Firefox浏览器中使用Web SDK，加入房间失败怎么办?**

  请排查Firefox浏览器的H264插件是否安装。浏览器中输入**about:addons**，跳转到插件安装页面，查看H264插件是否安装完成，如未安装请在该页面更新安装。

- **使用Mac Chrome浏览器屏幕分享失败，提示 "NotAllowedError: Permission denied by system" 或 "NotReadableError: Could not start video source" 时怎么办?**

  可能是由于未开启浏览器的屏幕录制权限导致。您可以通过在Mac的“设置 > 安全性与隐私 > 隐私 > 屏幕录制”中，打开Chrome屏幕录制授权后，重启Chrome浏览器。

8.7 修订记录

表 8-15 修订记录

修改时间	修改说明
2023-11-30	第十三次正式发布 本次变更如下： **客户端对象（Client）**新增旁路推流接口：startLiveStreaming、updateLiveStreaming和stopLiveStreaming。
2022-06-29	第十二次正式发布 本次变更如下： **主入口（HRTC）**createClient接口的countryCode入参修改为可选参数。
2022-06-21	第十一次正式发布 本次变更如下： • **客户端对象（Client）**新增setNetworkBandwidth接口。 • 优化部分文档描述。
2022-03-24	第十次正式发布 本次变更如下： 修改appid获取方式的相关描述。

修改时间	修改说明
2022-02-25	第九次正式发布 本次变更如下： • **主入口（HRTC）**createClient接口增加countryCode必选入参。客户端须修改代码，否则使用2.0.3 SDK创建客户端会失败。参数domain由必选修改为可选。 • **客户端对象（Client）**subscribe、unsubscribe接口增加autoAdjustResolution可选入参。 • **客户端对象（Client）**删除getLocalVideoStats、getRemoteVideoStats的入参。 • **客户端对象（Client）**新增接口：setProxyServer、setTurnServer、enableRtcStats。 • **客户端事件通知（ClientEvent）**新增事件：rtc-stats。 • **流对象（Stream）**getStreamInfo接口返回参数中新增RTCAudioProfile类型。 • **本地流对象（LocalStream）**addResolution接口新增audio可选入参。 • **客户端事件通知（ClientEvent）**删除事件：live-streaming-stopped、live-streaming-started。 • **流事件通知（RTCStreamEvent）**新增事件：audio-mixing-played、audio-mixing-finished。 • 新增**客户端错误码**。 • 优化部分文档描述。
2021-12-02	第八次正式发布 本次变更如下： • 增加常见问题列表及处理手段。 • 增加浏览器支持的列表及使用限制。 • 优化部分文档描述。
2021-11-22	第七次正式发布 本次变更如下： • 支持Firefox、Safari 11/12等浏览器适配。 • 优化部分文档描述。
2021-10-09	第六次正式发布 本次变更如下： • 支持Safari 13+浏览器适配。 • 支持移动端浏览器适配。 • 优化部分文档描述。

修改时间	修改说明
2021-06-04	第五次正式发布 本次变更如下： 增加双流能力，相关接口变更包括： • **流对象（Stream）**新增接口：getStreamInfo增加分辨率入参。 • **本地流对象（LocalStream）**新增接口：addResolution 和 removeResolution，增加分辨率入参。 • 支持端口收敛。 • 支持Unified-plan模式。 • getRemoteVideoStats接口返回数据变更：mainStream属性对应的值，修改为RemoteVideoStats[]。
2021-01-28	第四次正式发布 本次变更如下： • **客户端对象（Client）**新增接口：enableTopThreeAudioMode、setVolume4TopThree、muteAudio4TopThree、enableStreamStateDetection。 • **客户端事件通知（ClientEvent）**新增事件：stream-interrupted、stream-recovered。 • **本地流对象（LocalStream）**新增接口：bindScreenAudio2RelatedStream。 • **流事件通知（RTCStreamEvent）**新增事件：screen-sharing-stopped。
2020-12-25	第三次正式发布 本次变更如下： • 增加本地流LocalStream的混音相关接口。 • 增加客户端事件ClientEvent的设备列表变更相关事件。 • 优化部分文档描述。
2020-11-26	第二次正式发布 本次变更如下： • 新增TypeScript版本限制。 • 新增RTC **接入鉴权**方法说明。
2020-11-18	第一次正式发布

9 接入鉴权

为保证SparkRTC的通信安全，当用户加入房间时，华为云SparkRTC服务需要对其进行接入鉴权。本章节主要介绍华为云SparkRTC接入鉴权的实现原理及鉴权签名的生成方法。

鉴权原理

华为云SparkRTC系统使用数字签名作为接入鉴权方式，需要在SDK加入房间时设置“signature”和“ctime”。“signature”为标识签名，由租户使用华为云SparkRTC提供的“app_key”，“app_id”以及当前的“room_id”，“user_id”，“ctime”，按照华为SparkRTC的**签名生成样例**自行生成。具体参数说明请参见**表9-1**。

```
//认证用的app_key和app_id硬编码至代码中或以明文形式存储会有极大风险。建议密文形式配置存储在文件或者环境变量中，使用时解密，以确保安全。本例以app_key和app_id存放至环境变量为例，运行前请先在本地环境中设置完成环境变量APP_KEY和APP_ID。
app_key =  System.getenv("APP_KEY");
app_id =  System.getenv("APP_ID");
signature = hmacSha256(app_key,(app_id + room_id + user_id + ctime))
```

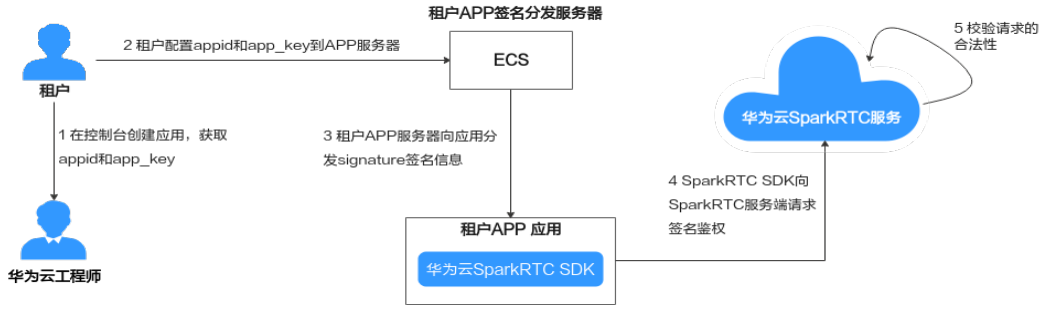
表 9-1 参数说明

参数	说明
app_key	华为云SparkRTC针对每个app生成的鉴权密钥，需要安全保存，谨防泄漏。 app_key的获取方法请参见**如何获取密钥？**。
app_id	华为云SparkRTC生成的应用ID。 app_id请在**实时音视频控制台**的“应用管理”中获取。
room_id	租户自行创建的房间ID。
user_id	租户接入华为云SparkRTC系统的用户ID。

参数	说明
ctime	签名鉴权的过期时间。是系统当前UTC时间（unix时间戳）加上鉴权过期时间（推荐2小时，最长需要小于12小时）。单位为秒。 **说明** ctime为创建时间+过期时间，例如，当前时间为9点，鉴权过期时间为30分钟，则ctime为9点30分。即超过9点30分后，signature签名将失效。

建议租户构建自己的应用签名分发服务器，以防止“app_key”下沉到终端APP的过程中造成不必要的泄漏，鉴权原理如图9-1所示。

图 9-1 鉴权原理



签名生成方法

您可以参考如下方法生成对应的签名。

1. 将“app_id”、“room_id”，“user_id ”和“ctime”拼接为一个字符串。

```
long ctime = System.currentTimeMillis() / 1000 + 60 * 60;  //有效时间为1小时，单位是秒
String content = app_id + "+" + room_id + "+" + user_id + "+" + ctime;
```

2. 使用“app_key”，通过HMAC-SHA256方式将字符串“content”进行加密，得到签名字符串。

```
String signatureStr = hmacSha(appKey, content, "HmacSHA256");
static String hmacSha(String KEY, String VALUE, String SHA_TYPE) {
    try {
        SecretKeySpec signingKey = new SecretKeySpec(KEY.getBytes("UTF-8"), SHA_TYPE);
        Mac mac = Mac.getInstance(SHA_TYPE);
        mac.init(signingKey);
        byte[] rawHmac = mac.doFinal(VALUE.getBytes("UTF-8"));

        byte[] hexArray = {
                (byte) '0', (byte) '1', (byte) '2', (byte) '3',
                (byte) '4', (byte) '5', (byte) '6', (byte) '7',
                (byte) '8', (byte) '9', (byte) 'a', (byte) 'b',
                (byte) 'c', (byte) 'd', (byte) 'e', (byte) 'f'
        };
        byte[] hexChars = new byte[rawHmac.length * 2];
        for (int j = 0; j < rawHmac.length; j++) {
            int v = rawHmac[j] & 0xFF;
            hexChars[j * 2] = hexArray[v >>> 4];
            hexChars[j * 2 + 1] = hexArray[v & 0x0F];
        }
        return new String(hexChars);
    } catch (Exception ex) {
        throw new RuntimeException(ex);
```

```
    }
  }
```

签名生成样例

为防止“app_key”密钥泄漏，建议您配置自己的应用签名分发服务器，向服务器传入“app_id”、“room_id”，“user_id”和“ctime”后，由服务器返回签名。详细代码示例如下所示：

```
package com.xxx.xxx.utils;

import androidx.annotation.NonNull;

import com.alibaba.fastjson.JSON;
import com.huawei.rtcdemo.Constants;

import java.io.IOException;

import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;

import okhttp3.Response;

public class SignatureUtil {
    private static final String TAG = "SignatureUtil";

    public interface onSignatureSuccess {
        void onSuccess(String signature);
    }

    public static void getSignature(String appid, String roomid, String userid, long ctime, String signatureKey,
onSignatureSuccess callback) {
        if (Constants.RTC_SIGNATURE_USE_LOCAL) {
            getSignatureLocal(appid, roomid, userid, ctime, signatureKey, callback);
        } else {
            getSignatureRemote(appid, roomid, userid, ctime, callback);
        }
    }

    private static void getSignatureLocal(String appid, String roomid, String userid, long ctime, String
signatureKey, @NonNull onSignatureSuccess callback) {
        String content = appid + "+" + roomid + "+" + userid + "+" + ctime; // here "+" is real char in content.
        String signature = SignatureUtil.hmacSha256(signatureKey, content);
        callback.onSuccess(signature);
    }

    private static void getSignatureRemote(String appid, String roomid, String userid, long ctime, @NonNull
onSignatureSuccess callback) {
        new Thread(new Runnable() {
            @Override
            public void run() {
                HttpUtil httpUtil = new HttpUtil();
                // Constants.RTC_SIGNATURE_URL：带用户自己应用签名的分发服务器地址
                String url = Constants.RTC_SIGNATURE_URL + "?appid=" + appid + "&roomid=" + roomid +
"&userid=" + userid + "&ctime=" + ctime;
                Response response = httpUtil.sendGetMethodWithHead(url, "X-AUTH-TOKEN",
Constants.RTC_AUTH_TOKEN);
                if (response == null) {
                    return;
                }

                if (response.isSuccessful()) {
                    try {
                        String json = response.body().string();
                        String signature = JSON.parseObject(json).getString("signature");
                        callback.onSuccess(signature);
                    } catch (IOException e) {
                        LogUtil.e(TAG, "getSignature failed:" + e.getMessage());
```

```
                }
            } else {
                LogUtil.e(TAG, "getSignature failed!");
            }
        }
    }).start();
}

public static String hmacSha256(String KEY, String VALUE) {
    return hmacSha(KEY, VALUE, "HmacSHA256");
}

private static String hmacSha(String KEY, String VALUE, String SHA_TYPE) {
    try {
        SecretKeySpec signingKey = new SecretKeySpec(KEY.getBytes("UTF-8"), SHA_TYPE);
        Mac mac = Mac.getInstance(SHA_TYPE);
        mac.init(signingKey);
        byte[] rawHmac = mac.doFinal(VALUE.getBytes("UTF-8"));

        byte[] hexArray = {
                (byte) '0', (byte) '1', (byte) '2', (byte) '3',
                (byte) '4', (byte) '5', (byte) '6', (byte) '7',
                (byte) '8', (byte) '9', (byte) 'a', (byte) 'b',
                (byte) 'c', (byte) 'd', (byte) 'e', (byte) 'f'
        };
        byte[] hexChars = new byte[rawHmac.length * 2];
        for (int j = 0; j < rawHmac.length; j++) {
            int v = rawHmac[j] & 0xFF;
            hexChars[j * 2] = hexArray[v >>> 4];
            hexChars[j * 2 + 1] = hexArray[v & 0x0F];
        }
        return new String(hexChars);
    } catch (Exception ex) {
        throw new RuntimeException(ex);
    }
}
}
```

10 附录

10.1 Grs 国家/地区码对照表

DR1：中国区

国家/地区名（中文全称）	国家/地区名（英文全称）	国家/地区码（英文缩写）
中国	China	CN

DR2：亚非拉（新加坡）

国家/地区名（中文全称）	国家/地区名（英文全称）	国家/地区码（英文缩写）
阿联酋	The United Arab Emirates	AE
阿富汗	Afghanistan	AF
安提瓜和巴布达	Antigua And Barbuda	AG
安圭拉	Anguilla	AI
亚美尼亚	Armenia	AM
安哥拉	Angola	AO
南极洲	Antarctica	AQ
阿根廷	Argentina	AR
美属萨摩亚	American Samoa	AS
阿鲁巴	Aruba	AW
阿塞拜疆	Azerbaijan	AZ
巴巴多斯	Barbados	BB

国家/地区名（中文全称）	国家/地区名（英文全称）	国家/地区码（英文缩写）
孟加拉	Bengal	BD
布基纳法索	Burkina Faso	BF
巴林	Bahrain	BH
布隆迪	Burundi	BI
贝宁	Benin	BJ
圣巴泰勒米岛	Saint Barthélemy	BL
百慕大	Bermuda	BM
文莱	Brunei Darussalam	BN
玻利维亚	Bolivia	BO
巴西	Brazil	BR
巴哈马	The Bahamas	BS
不丹	Bhutan	BT
布韦岛	Bouvet Island	BV
博茨瓦纳	Botswana	BW
伯利兹	Belize	BZ
科科斯群岛	Cocos (Keeling) Islands	CC
刚果金	Congo (Democratic Republic of the)	CD
中非	Central African Republic	CF
刚果布	Congo	CG
科特迪瓦	Côte d'Ivoire	CI
库克群岛	Cook Islands	CK
智利	Chile	CL
喀麦隆	Cameroon	CM
哥伦比亚	Colombia	CO
哥斯达黎加	Costa Rica	CR
佛特角	Cabo Verde	CV
圣诞岛	Christmas Island	CX
吉布提	Djibouti	DJ
多米尼克	Dominica	DM

国家/地区名（中文全称）	国家/地区名（英文全称）	国家/地区码（英文缩写）
多米尼加	Dominican Republic	DO
阿尔及利亚	Algeria	DZ
厄瓜多尔	Ecuador	EC
埃及	Egypt	EG
西撒哈拉	Western Sahara	EH
厄尔特里亚	Eritrea	ER
埃塞俄比亚	Ethiopia	ET
斐济群岛	Fiji	FJ
马尔维纳斯群岛（福克兰）	Falkland Islands	FK
密克罗尼西亚	Micronesia (Federated States of)	FM
加蓬	Gabon	GA
格林纳达	Grenada	GD
格鲁吉亚	Georgia	GE
法属圭亚那	French Guiana	GF
加纳	Ghana	GH
冈比亚	Gambia	GM
几内亚	Guinea	GN
瓜德罗普	Guadeloupe	GP
赤道几内亚	Equatorial Guinea	GQ
南乔治亚岛和南桑威奇群岛	South Georgia and the South Sandwich Islands	GS
危地马拉	Guatemala	GT
关岛	Guam	GU
几内亚比绍	Guinea-Bissau	GW
圭亚那	Guyana	GY
香港	Hong Kong	HK
赫德岛和麦克唐纳群岛	Heard Island and McDonald Islands	HM
洪都拉斯	Honduras	HN
海地	Haiti	HT

国家/地区名（中文全称）	国家/地区名（英文全称）	国家/地区码（英文缩写）
印度尼西亚	Indonesia	ID
印度	India	IN
英属印度洋领地	British Indian Ocean Territory	IO
伊拉克	Iraq	IQ
牙买加	Jamaica	JM
约旦	Jordan	JO
肯尼亚	Kenya	KE
吉尔吉斯斯坦	Kyrgyzstan	KG
柬埔寨	Cambodia	KH
基里巴斯	Kiribati	KI
科摩罗	Comoros	KM
圣基茨和尼维斯	Saint Kitts and Nevis	KN
科威特	Kuwait	KW
开曼群岛	Cayman Islands	KY
哈萨克斯坦	Kazakhstan	KZ
老挝	Laos	LA
黎巴嫩	Lebanon	LB
圣卢西亚	Saint Lucia	LC
斯里兰卡	Sri Lanka	LK
利比里亚	Liberia	LR
莱索托	Lesotho	LS
利比亚	Libya	LY
摩洛哥	Morocco	MA
马达加斯加	Madagascar	MG
马绍尔群岛	Marshall islands	MH
马里	Mali	ML
缅甸	Myanmar	MM
蒙古	Mongolia	MN
澳门	Macao	MO

国家/地区名（中文全称）	国家/地区名（英文全称）	国家/地区码（英文缩写）
北马里亚纳群岛	Northern Mariana Islands	MP
马提尼克	Martinique	MQ
毛里塔尼亚	Mauritania	MR
蒙特塞拉特岛	Montserrat	MS
毛里求斯	Mauritius	MU
马尔代夫	Maldives	MV
马拉维	Malawi	MW
墨西哥	Mexico	MX
马来西亚	Malaysia	MY
莫桑比克	Mozambique	MZ
纳米比亚	Namibia	NA
新喀里多尼亚	New Caledonia	NC
尼日尔	Niger	NE
诺福克岛	Norfolk Island	NF
尼日利亚	Nigeria	NG
尼加拉瓜	Nicaragua	NI
尼泊尔	Nepal	NP
瑙鲁	Nauru	NR
纽埃	Niue	NU
阿曼	Oman	OM
巴拿马	Panama	PA
秘鲁	Peru	PE
法属波利尼西亚	French Polynesia	PF
巴布亚新几内亚	Papua New Guinea	PG
菲律宾	Philippines	PH
巴基斯坦	Pakistan	PK
皮特凯恩群岛	Pitcairn	PN
波多黎各	Puerto Rico	PR
巴勒斯坦	Palestine, State of	PS

国家/地区名（中文全称）	国家/地区名（英文全称）	国家/地区码（英文缩写）
帕劳	Palau	PW
巴拉圭	Paraguay	PY
卡塔尔	Qatar	QA
留尼汪	Reunion !Réunion	RE
卢旺达	Rwanda	RW
沙特阿拉伯	Saudi Arabia	SA
所罗门群岛	Solomon Islands	SB
塞舌尔	Seychelles	SC
新加坡	Singapore	SG
圣赫勒拿	Saint Helena, Ascension and Tristan da Cunha	SH
塞拉利昂	Sierra Leone	SL
塞内加尔	Senegal	SN
索马里	Somalia	SO
苏里南	Suriname	SR
南苏丹	South Sudan	SS
圣多美和普林西比	Sao Tome and Principe	ST
萨尔瓦多	El Salvador	SV
斯威士兰	Swaziland	SZ
特克斯和凯科斯群岛	Turks and Caicos Islands	TC
乍得	Chad	TD
法属南部领地	French Southern Territories	TF
多哥	Togo	TG
泰国	Thailand	TH
塔吉克斯坦	Tajikistan	TJ
托克劳	Tokelau	TK
东帝汶	Timor-Leste	TL
土库曼斯坦	Turkmenistan	TM
突尼斯	Tunisia	TN
汤加	Tonga	TO

国家/地区名（中文全称）	国家/地区名（英文全称）	国家/地区码（英文缩写）
特立尼达和多巴哥	Trinidad and Tobago	TT
图瓦卢	Tuvalu	TV
台湾	Taiwan, Province of China	TW
坦桑尼亚	Tanzania, United Republic of	TZ
乌干达	Uganda	UG
乌拉圭	Uruguay	UY
乌兹别克斯坦	Uzbekistan	UZ
委内瑞拉	Venezuela (Bolivarian Republic of)	VE
英属维尔京群岛	Virgin Islands, British	VG
美属维尔京群岛	Virgin Islands, U.S.	VI
越南	Vietnam	VN
瓦努阿图	Vanuatu	VU
瓦利斯和富图纳	Wallis and Futuna	WF
萨摩亚	Samoa	WS
也门	Yemen	YE
马约特	Mayotte	YT
南非	South Africa	ZA
赞比亚	Zambia	ZM
津巴布韦	Zimbabwe	ZW
古巴	Cuba	CU
朝鲜	North Korea	KP
苏丹	Sudan	SD
叙利亚	Syria	SY
日本	Japan	JP
韩国	Korea (Republic of)	KR

DR3：欧洲区（德国）

国家/地区名（中文全称）	国家/地区名（英文全称）	国家/地区码（英文缩写）
安道尔	Andorra	AD
阿尔巴尼亚	Albania	AL
奥地利	Austria	AT
奥兰群岛	Aland Islands	AX
波黑	Bosnia and Herzegovina	BA
比利时	Belgium	BE
保加利亚	Bulgaria	BG
加拿大	Canada	CA
瑞士	SWITZERLAND	CH
塞浦路斯	Cyprus	CY
捷克	Czech Republic	CZ
德国	Germany	DE
丹麦	Denmark	DK
爱沙尼亚	Estonia	EE
西班牙	Spain	ES
芬兰	Finland	FI
法罗群岛	Faroe Islands	FO
法国	France	FR
英国	United Kingdom of Great Britain and Northern Ireland	GB
根西岛	Guernsey	GG
直布罗陀	Gibraltar	GI
格陵兰	Greenland	GL
希腊	Greece	GR
克罗地亚	Croatia	HR
匈牙利	Hungary	HU
爱尔兰	Ireland	IE
以色列	Israel	IL
马恩岛	Isle of Man	IM

国家/地区名（中文全称）	国家/地区名（英文全称）	国家/地区码（英文缩写）
冰岛	Iceland	IS
意大利	Italy	IT
泽西岛	Jersey	JE
列支敦士登	Liechtenstein	LI
立陶宛	Lithuania	LT
卢森堡	Luxembourg	LU
拉脱维亚	Latvia	LV
摩纳哥	Monaco	MC
摩尔多瓦	Moldova	MD
黑山	Montenegro	ME
圣马丁	Saint Martin (French part)	MF
马其顿	Macedonia (the former Yugoslav Republic of)	MK
马耳他	Malta	MT
荷兰	Netherlands	NL
挪威	Norway	NO
波兰	Poland	PL
圣皮埃尔和密克隆	Saint Pierre and Miquelon	PM
葡萄牙	Portugal	PT
罗马尼亚	Romania	RO
塞尔维亚	Serbia	RS
瑞典	Sweden	SE
斯洛文尼亚	Slovenia	SI
斯瓦尔巴群岛和扬马延岛	SJMSvalbard	SJ
斯洛伐克	Slovakia	SK
圣马力诺	San Marino	SM
圣马丁岛	Sint Maarten (Dutch part)	SX
土耳其	Türkiye	TR

国家/地区名（中文全称）	国家/地区名（英文全称）	国家/地区码（英文缩写）
美国本土外小岛屿	United States Minor Outlying Islands	UM
美国	America	US
梵蒂冈	Holy See	VA
圣文森特和格林纳丁斯	Saint Vincent and the Grenadines	VC
科索沃	Kosovo	XK->YK
澳大利亚	Australia	AU
新西兰	New Zealand	NZ
荷属安的列斯	Netherlands Antilles	BQ->AN
乌克兰	Ukraine	UA
库拉索	Curaçao	CW

11 修订记录

表 11-1 修订记录

发布日期	修改说明
最新时间	实时音视频各端SDK的修订记录，详见各端节点下的“修订记录”。
2020-03-30	第一次正式商用发布。